

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9429659

**CLARE: A computer-supported collaborative learning
environment based on the thematic structure of scientific text**

Wan, Dadong, Ph.D.

University of Hawaii, 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

CLARE: A COMPUTER-SUPPORTED COLLABORATIVE LEARNING
ENVIRONMENT BASED ON THE THEMATIC
STRUCTURE OF SCIENTIFIC TEXT

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMMUNICATION & INFORMATION SCIENCES

MAY 1994

By

Dadong Wan

Dissertation Committee:

Philip M. Johnson, Chairperson
Daniel Watanabe
Martha Crosby
William Remus
Steve Goldberg

© Copyright 1994

by

Dadong Wan

To
My Parents

Acknowledgments

Dissertation research is a solitary journey. As a researcher in collaborative work, I also view it as a joint effort. The work reported here, at least, would not have reached the present point without support, discussions, and other forms of *collaboration* with many people, a few of whom are mentioned below.

Philip Johnson, my advisor, provided me the opportunity and the environment for the past three years. I thank him for taking a chance on me, for his intellectual, financial, and emotional support, for the time he spent reviewing my code and commenting drafts of my dissertation chapters, and for his vision of making his group world-class in research and development.

I am grateful to other members of my committee for their guidance and input. Daniel Watanabe raised a number of insightful questions during both my proposal and final defenses. Martha Crosby increased my awareness of cognitive and behavioral issues related to CLARE. Bill Remus made several suggestions on improving the CLARE experiment. Steve Goldberg was a source of immense encouragement. I thank Steve for his enthusiasm for what I was doing, and for his perspective on how CLARE might be used in ways that had not yet occurred to me.

I would like to express my special thanks to Jintae Lee for several valuable discussions we had about CLARE and for his insights on such issues as marketability and parsimonious representation. I also thank him for encouraging his students to use CLARE.

I feel fortunate for being part of a supportive group while working on this project. I thank both current and former members of the Collaborative Software Development Laboratory (CSDL): Danu Tjahjono, Robert Brewer, Cam Moore, Rosemary Andrada, Carol Izaki, Bruce Campbell, and Ram Hariharan. Their contribution ranges from helping refine my research ideas, reviewing my design and code, to willingly serving as guinea pigs at various stages of the CLARE prototype.

I am grateful to a number of friends who have strengthened and challenged me at difficult moments of my life: Andres Edwards, Yong Ma, Kexian Hu, Dave Kim, Spike Yamamoto, and Scotty Blaisdell. Special thanks go to Brock and Anne Roby, whose convictions on what they believe in often reminded me what ultimately matters in life.

Finally, I thank my parents for their understanding of me not writing to them for months and not visiting them for years. I wish I had devoted myself to them as much as I did to this work. On the other hand, I know deeply in my heart that, even if I had, I could still not, nor will I ever be able to, pay back for what they have given to me. For that reason, I dedicate this dissertation to them.

Abstract

This dissertation presents a computer-based collaborative learning environment, called CLARE, that is based on the theory of learning as collaborative knowledge building. It addresses the question, “what can a computer do for a group of learners beyond helping them share information?” CLARE differs from virtual classrooms and hypermedia systems in three ways. First, CLARE is grounded on the theory of meaningful learning, which focuses the role of meta-knowledge in human learning. Instead of merely allowing learners to share information, CLARE provides an explicit meta-cognitive framework, called RESRA, to help learners interpret information and build knowledge. Second, CLARE defines a new group process, called SECAI, that guides learners to systematically analyze, relate, and discuss scientific text through a set of structured steps. *summarization, evaluation, comparison, argumentation, and integration*. Third, CLARE provides a fine-grained, non-obtrusive instrumentation mechanism that keeps track of the usage process of its users. Such data forms an important source of feedback for enhancing the system and a basis for rigorously studying collaboration learning behaviors of CLARE users.

CLARE was evaluated through sixteen usage sessions involving six groups of students from two classes. The experiments consist of a total of about 300 hours of usage and over 80,000 timestamps. The survey shows that about 70% of learners think that CLARE provides a novel way of understanding scientific text, and about 80% of learners think that CLARE provides a novel way of understanding their peers’ perspectives. The analysis of the CLARE database and the process data also reveals that learners differ greatly in their interpretations of RESRA, strategies for comprehending the online text, and understanding of the selected artifact. It is also found that, despite the large amount of time spent on summarization (up to 66%), these learners often fail to correctly represent important features of scientific text and the relationships between those features. Implications of these findings at the design, empirical, and pedagogical levels are discussed.

Table of Contents

Acknowledgments	v
Abstract	vii
List of Tables	xiv
List of Figures	xvii
1 Overview	1
1.1 Collaborative learning using CLARE	3
1.1.1 A usage scenario	3
1.1.2 Discussion	10
1.2 Motivation	12
1.2.1 Technological biases in current learning support systems	12
1.2.2 Cognitive learning theory and concept mapping	13
1.2.3 Toward a theory-based collaborative learning support environment	15
1.3 Research thesis	15
1.4 Research contributions	16
1.5 Scope and limitations	18
1.6 Organization of this dissertation	18
2 Toward a Representation-Based Approach to Collaborative Learning from Scientific Text	21
2.1 From semantic nets to concept maps: knowledge representation in human learning	22
2.1.1 Two types of knowledge representation	22

	2.1.2	Content learning, meta-learning, and knowledge representation	26
	2.1.3	Cognitive learning theory and concept mapping	28
	2.1.4	Critiques on concept maps: a KR perspective	29
	2.2	Representation of scientific text	32
	2.2.1	The role of scientific text	33
	2.2.2	The structure of scientific text	35
	2.2.3	RESRA: a KR scheme for representing scientific text	36
	2.3	Toward a representation-based model of collaborative learning	39
	2.3.1	Collaborative learning from scientific text	39
	2.3.2	SECAI: the collaborative learning model	40
	2.3.3	The role of the RESRA representation	42
	2.4	Summary and conclusions	43
3		RESRA: the Representation Language	44
	3.1	Representational requirements	44
	3.2	Basic concepts	46
	3.3	RESRA primitives	47
	3.3.1	Summarative primitives	47
	3.3.2	Evaluative primitives	56
	3.3.3	Argumentative primitives	59
	3.3.4	Integrative primitives	59
	3.4	Canonical RESRA Forms (CRFs)	62
	3.4.1	Overview	62
	3.4.2	Five example CRFs	63
	3.5	Extending RESRA	74
4		CLARE: the System	77

4.1	Basic requirements	77
4.1.1	Conceptual requirements	78
4.1.2	Data requirements	81
4.1.3	Usability requirements	81
4.2	Main design considerations	82
4.2.1	Layered + object-oriented design	82
4.2.2	“Services” versus “interface”	83
4.3	Architecture	84
4.3.1	Kernel	84
4.3.2	Preparation	85
4.3.3	Exploration	86
4.3.4	Consolidation	87
4.3.5	Utilities	88
4.4	Interface	88
4.5	The road map	91
4.6	History and status	91
4.6.1	Brief history	91
4.6.2	Current status	95
5	Hypotheses and Experiments	97
5.1	The research problem revisited	97
5.2	Hypotheses	100
5.2.1	RESRA	100
5.2.2	SECAI	102
5.2.3	CLARE	103
5.3	Data collection	103

	5.3.1	Outcome data	103
	5.3.2	Process data	107
	5.3.3	Assessment data	108
	5.3.4	Hypotheses and data collection	108
	5.4	Data analyses	115
	5.5	Experiments	115
	5.5.1	Task	115
	5.5.2	Subjects	117
	5.5.3	Procedures	117
	5.5.4	Execution Plan	118
6		Evaluation Results	121
	6.1	Summary of the experiments	121
	6.2	Overview of the results	124
	6.3	Hypothesis evaluation	128
	6.3.1	RESRA	128
	6.3.2	SECAI	132
	6.3.3	CLARE	132
	6.4	Issues on the RESRA representation	139
	6.4.1	Categorizing representational differences	139
	6.4.2	Common errors in using RESRA	144
	6.4.3	Implications	150
	6.5	CLARE usage strategies	150
	6.5.1	Navigation	151
	6.5.2	Summarization	152
	6.5.3	Evaluation	154

	6.6	Collaborative learning using CLARE: a case analysis	155
	6.6.1	Overview of the CLARE session	156
	6.6.2	Analysis of the exploration result	158
	6.6.3	Analysis of the consolidation result	168
	6.6.4	Discussion	172
	6.7	Summary and conclusions	174
7		Related Work	178
	7.1	Theoretical underpinnings	178
	7.1.1	Constructionism	179
	7.1.2	Assimilation theory of cognitive learning	180
	7.2	Representation and human learning	181
	7.2.1	Schema theory, representation, and learning	181
	7.2.2	RESRA and other representation schemes	183
	7.3	Collaborative learning systems: technologies and outcomes	186
	7.3.1	Virtual classroom systems	187
	7.3.2	Collaborative writing systems	189
	7.3.3	Hypermedia systems	190
	7.3.4	Collaborative knowledge-building tools	192
	7.4	Summary	195
8		Conclusions	196
	8.1	A RESRA representation of this dissertation	196
	8.2	Main Contributions	199
	8.2.1	Collaborative learning from scientific text	200
	8.2.2	RESRA representation language	202
	8.2.3	Design and implementation of the CLARE system	203

	8.2.4	Empirical evaluation of CLARE	204
	8.3	Where CLARE is heading	205
	8.3.1	Short-term goals	205
	8.3.2	Long-term directions	210
A		CLARE Assessment Questionnaire	216
B		CLARE Feedback Questionnaire	220
C		A CLARE Session Summary	222
	C.1	Scott	222
	C.2	Chris	227
	C.3	Mary	232
	C.4	Todd	238
	C.5	Argumentation nodes	242
	C.6	RESRA tuples	243
		Bibliography	246

List of Tables

2.1	KR schemes used by human learners and AI programs	25
2.2	Characteristics of content and meta-learning	27
2.3	A synopsis of RESRA primitive node types	38
2.4	Five principal activities of collaborative learning from scientific text	41
5.1	A synopsis of the ten hypotheses on CLARE	104
5.2	A synopsis of data to be gathered	120
6.1	Summary statistics on CLARE experiments	123
6.2	Summary of major CLARE experimental findings	125
6.3	Comparison of 16 learner's summarizations of [Fagan76] with the <i>base representation</i>	141
6.4	Distributions of the number of <i>rounds</i> on source nodes	151
6.5	Distribution of summarization strategies	154
6.6	Distribution of evaluation strategies	155
6.7	Distribution of summarative nodes by Scott, Chris, Mary, and Todd	159
6.8	Distribution of argumentative nodes by Scott, Chris, Mary, and Todd	168
C.1	Summarative nodes (EX) at a glance	245
C.2	Evaluative nodes (EX) at a glance	245
C.3	Summarative nodes (CON) at a glance	245
C.4	Evaluative nodes (CON) at a glance	245

List of Figures

1.1	An example of Scott's view during the exploration phase	4
1.2	Scott's representation of [KTBB92] (condensed)	5
1.3	Chris' representation of [KTBB92] (condensed)	6
1.4	A comparative view of problem instances by Scott Chris, Mary, and Todd	7
1.5	Collaborative knowledge-building in CLARE	11
2.1	Knowledge representation in human learning and AI contexts	24
2.2	Relationships between knowledge representation, meta-learning, and content learning	28
2.3	An example concept map on knowledge construction and acquisition from the perspective of the assimilation theory of cognitive learning (from [NG84])	30
2.4	Concept maps versus semantic nets: an example	31
2.5	Collaborative learning from scientific text	42
2.6	The role of RESRA in collaborative learning	43
3.1	RESRA summarative primitives at a glance	48
3.2	RESRA evaluative primitives at a glance	57
3.3	RESRA argumentative primitives at a glance	60
3.4	A CRF for concept papers	64
3.5	A RESRA representation of [JT92]	65
3.6	A CRF for experience papers	66
3.7	A RESRA representation of [Bush, 90]	67

3.8	A CRF for empirical papers	68
3.9	An example RESRA representation of [CSM79]	70
3.10	A CRF for research essays	71
3.11	A RESRA representation of [Knuth92]	72
3.12	A CRF for survey papers	73
3.13	A RESRA representation of [Tichy, 92]	75
4.1	CLARE architecture	85
4.2	Example view of CLARE during exploration	89
4.3	Example view of CLARE during consolidation	90
4.4	CLARE functional road map	92
4.5	Evolutionary path of CLARE	93
6.1	Distribution of average CLARE usage time during exploration and consolidation	124
6.2	How many times did you find important contents and relationships cannot be expressed in RESRA node and link primitives, respectively?	127
6.3	How many times did CRFs sensitize you about ambiguities and gaps in the paper?	131
6.4	How often should CLARE be used in classroom settings?	133
6.5	Would you recommend using CLARE for studying research papers?	134
6.6	Learners' ranking of the usefulness of CLARE functions	136
6.7	Barriers to effective use of CLARE	137
6.8	A RESRA representation of major themes of [Fagan76]	140
6.9	The source node structure of the CLARE database	152
6.10	A RESRA representation of [FD91]	157
6.11	Distribution of the CLARE usage time	158

6.12	A comparative view of four RESRA representations of [Flynn90]	159
6.13	Distribution of summarative nodes over the source nodes	160
6.14	Overview of summarization and argumentation by Scott, Chris, Mary, and Todd	169
7.1	IBIS model of argumentation (based on [Conklin87])	183
7.2	Toulmin's model of argumentation (based on [Toulmin52])	184
7.3	Novak and Gowin's Vee Diagram (from [NG84])	185
8.1	A RESRA representation of the major themes of this dissertation	199
8.2	CLARE as an environment for supporting collaborative knowledge-building	201

Chapter 1

Overview

This dissertation presents a new approach to collaborative learning that is based on the thematic structure of scientific text¹ and the theory of learning as collaborative knowledge-building. It also describes a software system called CLARE that embodies such a conceptual approach. Furthermore, it discusses the experience from sixteen usage sessions of CLARE by six groups of students from two different university-level classes. This usage indicates that CLARE is a useful environment to support meaningful learning in collaborative settings.

In general, this research addresses the question, “what can the computer do for a group of learners beyond helping them share information?” This question can be rephrased more specifically as, “how can the higher-level knowledge — the knowledge about (1) deep structures of individual scientific artifacts, (2) inter-relationships between artifacts, and (3) processes of knowledge-construction — be used to facilitate meaningful learning in group settings?” Alternatively, “what kinds of structural, process-level, and computational support are required to help learners reconstruct and evaluate the thematic features of scientific artifacts, compare different interpretations of those features, deliberate reasonings behind those interpretations, and integrate different points of view on the artifact?”

CLARE responds to the above questions with the following features:

- A multi-user, distributed environment for supporting artifact-based knowledge-building;

¹Throughout this dissertation, the term *scientific text* is used interchangeably with *scientific artifacts*, *research literature*, and *learning artifacts*; they refer to the written record of human knowledge. Examples of scientific text include discussion papers, journal articles, conference papers, monographs. Literary text, such as poetry, short stories, and novels, is excluded unless explicitly noted otherwise.

- A two-phase, five-step process model for helping learners organize their collaborative learning activities;
- A thematically-oriented representation language that serves as the principal integrative mechanism between learners and various types of learning activities;
- A shared, evolving knowledge base that captures not only the final product but also the process that leads to that product; and
- An unobtrusive instrumentation mechanism that collects fine-grained data about the process of collaboration and learning.

CLARE is built upon a representation language called RESRA, which serves the following roles in collaborative knowledge-construction:

- A meta-cognitive tool that highlights essential thematic features and the relationships between these features;
- An organizational tool that allows the learner to dynamically and incrementally integrate at a fine-grained level various types of research artifacts;
- A shared frame of reference that facilitates communication and argumentation among learners; and
- A tool for studying the norms governing the written communication of scientific knowledge.

This chapter provides an overview of the dissertation. It begins with a hypothetical usage scenario that illustrates the basic process and main features of collaborative learning under the CLARE environment. The subsequent section describes the main motivations behind the current work. Next, it introduces the main thesis, which is followed by the major research contributions made by this work. The scope and limitations are highlighted next. The chapter concludes with an overview of the organization for the remainder of this dissertation.

1.1 Collaborative learning using CLARE

1.1.1 A usage scenario

Scott, Chris, Mary, and Todd² are in a research seminar on computer-supported cooperative work (CSCW). Unlike traditional seminars in which interactions among participants take place largely in a face-to-face setting, this seminar is featured with a structured, distributed online environment called CLARE. Rather than receiving a stack of printed papers to read, students are asked to analyze and discuss selected research literature online. The example paper used in the current scenario is “Supporting collaborative software development with ConversationBuilder” by Kaplan, et al [KTBB92]. The full-text of the paper is available online in CLARE in a hypertext format: each node corresponds to a *semantic unit* or a section of the paper. Individual nodes are connected via links derived directly from the structure of the artifact. Hence, students may navigate the entire document by simply following these links.

The study session is organized into two phases: *exploration* and *consolidation*. During the exploration phase, individual students summarize in terms of predefined representation primitives, such as problem and concept, what they view as important features of the paper. They also evaluate those features and the relationships between them by making critiques, raising questions, and offering suggestions. During the consolidation phase, they compare the result of their individual summarizations and evaluations to discern ambiguities, differences, and similarities. When ambiguities and/or differences occur, they may question, critique, propose, or clarify by providing new information. In response to other learners’ questions, critiques, or suggestions, one may defend, deliberate, or amend one’s own positions. Toward the end, learners go back to the online artifacts they have created and connect together similar and related positions and explanations to produce a more coherent body of knowledge.

²The names of these four users are fictitious. However, this usage scenario itself is only *somewhat* hypothetical: it is based on the result from a manual experiment on RESRA — the CLARE’s representation language — during the early phase of this research. The experiment was not done under CLARE, because the system was not yet completed at that time.

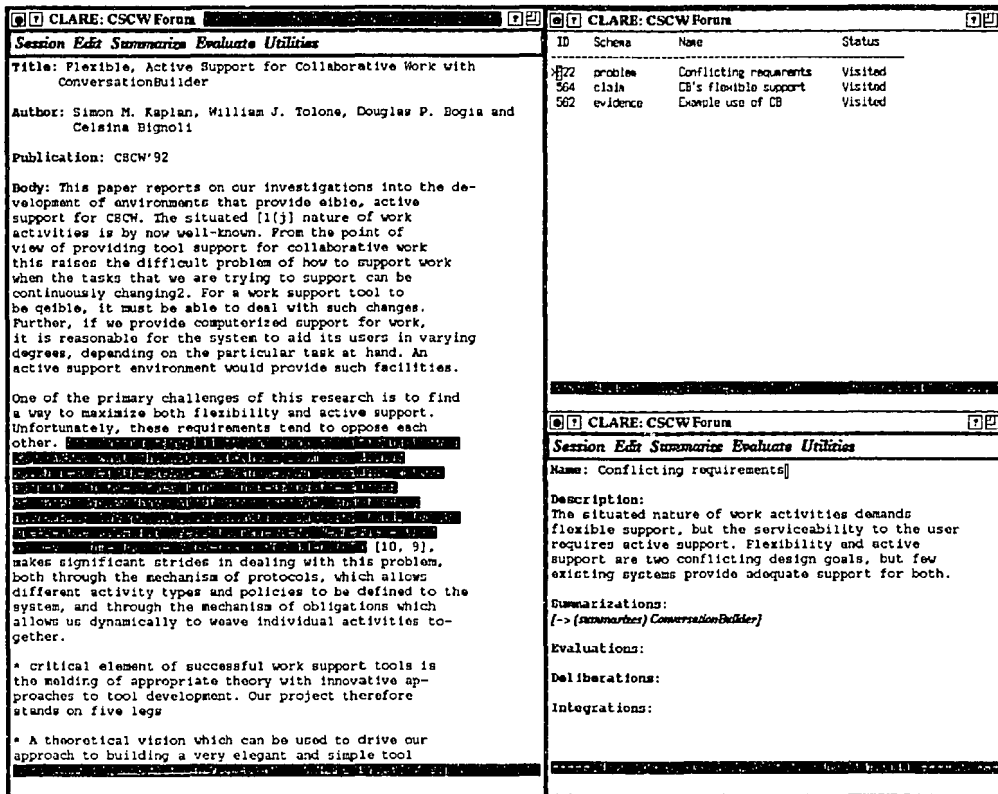


Figure 1.1: An example of Scott's view during the exploration phase

When Scott logged on to CLARE for the first time, he navigated through the entire document by following the built-in links to get a general sense of what the paper is about. He then zoomed-in to individual nodes and examined more closely their contents. When he encountered a paragraph or a text region which he thought was important, he highlighted it by dragging the mouse over it, and selected Create node from the popup menu. CLARE shows a submenu which lists all predefined node types, for example, `concept`, `claim`. Selecting any of these entries leads to the creation of a node of the corresponding type. For example, the left window in Figure 1.1 shows a highlighted region as describing the `problem` Scott considered that the author attempts to address in that paper. Hence, he created a `problem` node for this region, which is shown in the lower right window. The

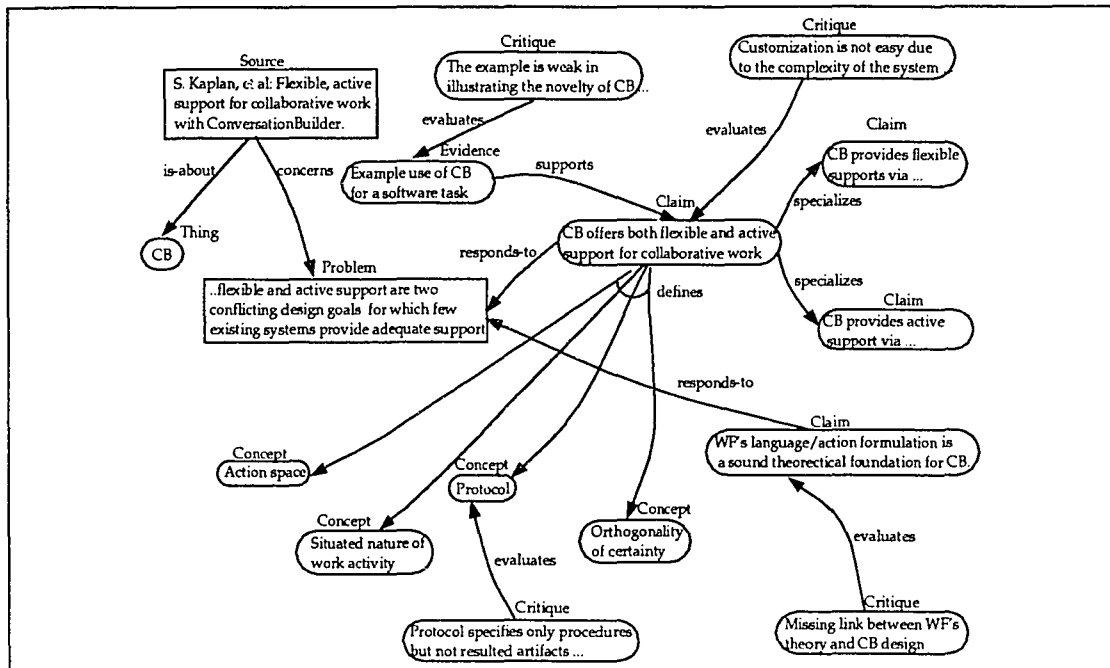


Figure 1.2: Scott's representation of [KTBB92] (condensed)

italicized text under the field **Summarizations**³ is the link to the region from which the node is derived. The main content field, namely, **Description**, contains Scott's description of what he thinks the authors attempt to convey. Note that the content of this field is not a verbatim copy of the original authors' statements. Rather, it articulates Scott's understanding of the selected text region and, to some extent, of the paper as a whole.

By following the same procedure, Scott created a **claim** node based on a different part of the paper, which states that ConversationBuilder offers a viable solution to the problem of active and flexible support. Because there seems to exist a direct relationship between the two nodes, Scott wanted to connect them together. Hence, he selected Enter LINK Mode from the Summarize menu, which creates two windows in the lower half of the screen. The two windows show the two nodes he has created thus far. Scott moves the mouse to the

³Summarizations here is a field label which means that all links underneath it are generated during summarization. The other three link fields, namely, **Evaluations**, **Deliberations**, and **Integrations** have the similar semantics.

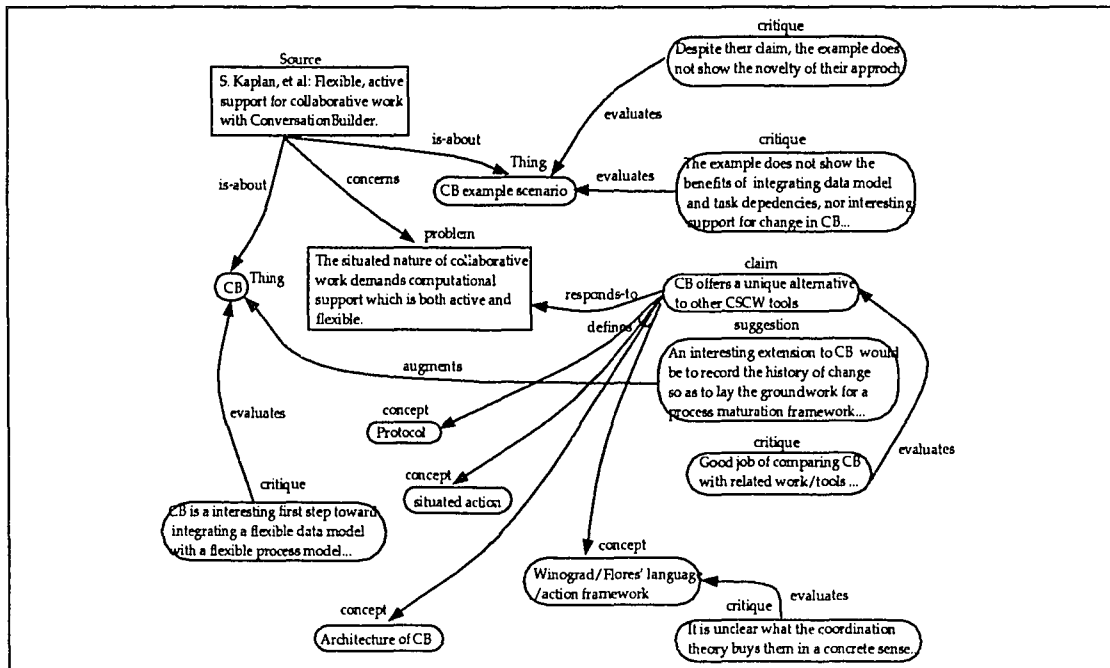


Figure 1.3: Chris' representation of [KTBB92] (condensed)

claim window, select *Responds-to PROBLEM* from the context-sensitive popup menu, followed by pressing the mouse on the **problem** window: a bidirectional link of the type *responds-to* was added between the two nodes. This link indicates that the claim is made with respect to to the **problem** or, alternatively, the **problem** is responded by that **claim**. Following this action, Scott exited from the link mode, and went on to create more nodes.

From reading the user guide [Wan93b], Scott learned that CLARE possesses some knowledge about the structure of the current paper. Hence, he wanted to know whether CLARE could provide him reasonable advice about what to look for next. He selected Consult for what's missing from the option Template Guide of the Summarize menu. CLARE pops up a new window, which shows the following information:

- The current paper is a *concept paper*⁴;
- Scott has thus far created one tuple (a pair of nodes connected by a link of a pre-

⁴CLARE characterizes the structural features of scientific text based on the artifact types. The *concept*

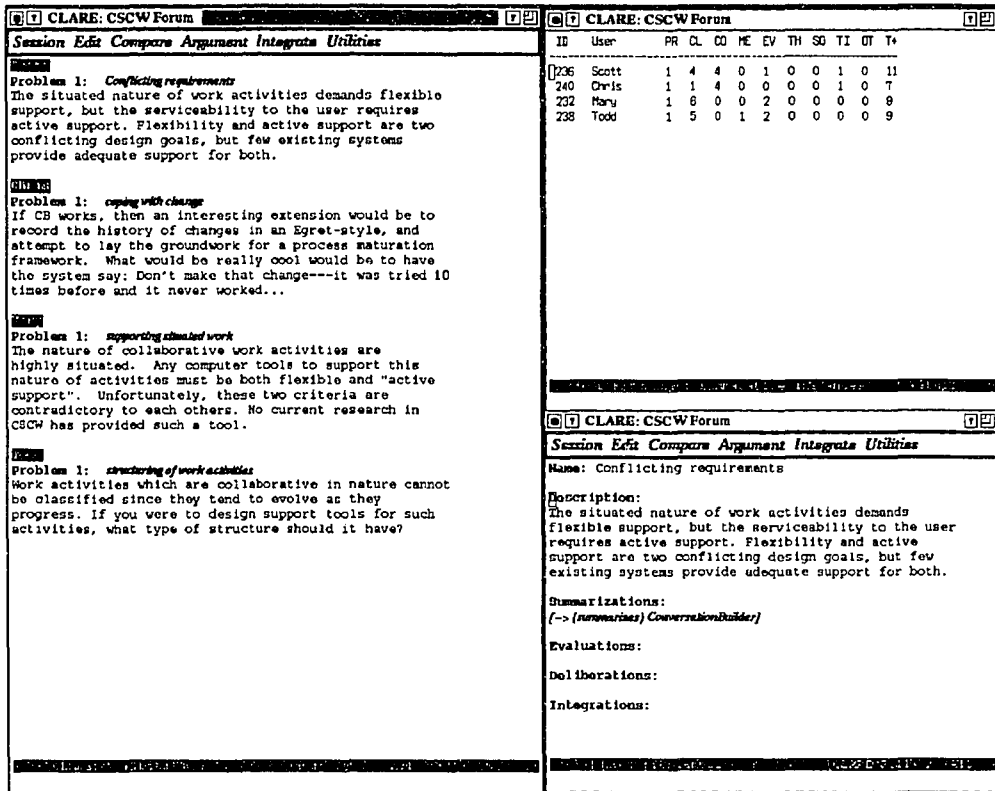


Figure 1.4: A comparative view of **problem** instances by Scott Chris, Mary, and Todd

defined type), namely, **claim** $\xrightarrow{\text{responds-to}}$ **problem**, and no orphan nodes, i.e., nodes which are not connected to other nodes; and

- Based on the structural knowledge the system knows about the current artifact, CLARE suggests Scott to consider creating tuple of the following types:

- **evidence** $\xrightarrow{\text{supports}}$ **claim**
- **claim** $\xrightarrow{\text{defines}}$ **concept**
- **claim** $\xrightarrow{\text{defines}}$ **method**

paper is one of five such predefined artifact types. Other types include *experience papers*, *empirical papers*, *essays*, and *survey papers*. See Section 3.4 for a detailed description of these artifact types.

Scott took to heart the above suggestions from CLARE; he went back to the source nodes one more time. Unlike previous passes in which he read to understand detailed contents, this time John was looking for specific themes, such as **evidence**, **concept**, and **method**. For example, he did discover an **evidence** from another section of the paper, namely, the example use of ConversationBuilder, which *supports* the **claim** he identified earlier. He also found a few important **concept** instances, such as “action space,” “protocol.” At the same time, he evaluated the paper content by creating a number of **critique** nodes. However, he was unable to find any **method** instance. Figure 1.2 shows an abbreviated, graphical depiction of Scott’s representation of [KTBB92]⁵.

While Scott was busy with his exploration of the paper, Chris, Mary, and Todd were also in and out of CLARE engaging in similar types of activities. During this phase, however, they could not see what each other were doing, that is, what text regions were highlighted, or what nodes and links were created. Rather, all four of them were independently deriving their own representations of the paper and their views on it. Figure 1.3, for example, shows an abbreviated depiction of Chris’ representation of the same paper. Note that Figure 1.2 and 1.3 are quite different not only in term of the type of nodes and links, the origins of these nodes but, perhaps most important of all, the contents of those nodes. These differences and their implications will become evident in the consolidation phase.

The consolidation phase was activated two days after the session began. This gave all four learners adequate time to complete the exploration phase. The primary goals of consolidation are:

- To expose the differences and ambiguities of individual learners’ interpretations and evaluations of the selected artifact;
- To deliberate, resolve, and augment these differences and ambiguities; and
- To link together similar and related views held by different learners.

⁵CLARE currently does not have a graphical browser. The graphical depiction here is paraphrased, abbreviated, and hand-drawn from the actual data.

Figure 1.4 shows a typical user view of CLARE during the consolidation phase. The left window presents primitive-based comparison of node instances created by Scott, Chris, Mary, and Todd. In the current example, the window shows the comparison of **problem** instances — four independent interpretations of what the original authors attempt to address in the paper. By displaying them next to each other, it is relatively easy to discern similarities, differences, and ambiguities. For example, although each learner has a **problem** node, their content is quite distinct: Scott’s and Mary’s characterizations are very similar; both explicitly mention the conflicting requirements between the situated nature of collaborative work that calls for flexible support, and the demand for active support. Todd’s statement, however, calls solely for flexible support. The latter is not intended by the original authors, for they explicitly state that providing *either flexible or active support* is not difficult; the difficulty only arises when both have to be satisfied. Chris’ **problem** introduces the issue of support for change, which is not touched on in the paper.

While CLARE’s comparison mode helps uncover differences and ambiguities, its argumentation feature supports deliberation and resolution of those differences and ambiguities. In the above example, for instance, although Chris’ view does not reflect that of the original authors, he did introduce some new themes to the scene, most noticeably, “EGRET” and “process maturation.” Since both Chris and Scott were involved in the design of EGRET system and interested in the process maturation, they could elaborate these themes by creating a separate **thing** node for EGRET, and a **concept** for process maturation. Similarly, Scott might request Todd to explain why he thought that the sole support for flexibility is a **problem** by creating a **question** node.

Finally, similar and related nodes can be integrated by selecting appropriate options from the Integrate menu. For example, since Scott’s and Mary’s **problem** nodes are similar, a *is-similar-to* link could be added between those two nodes by selecting the option Declare two nodes as SIMILAR from the Integrate menu. In addition, Mary believed that Scott’s representation was more articulated than hers, even though they both captured the same information. Hence, she decided to endorse Scott’s view by selecting Endorse current node from the Integrate menu.

One main result of the above comparative, argumentative, and integrative activities is a

deeper understanding of the content of the selected paper by Chris, Scott, Mary, and Todd. In addition, this process also leads to a knowledge base that captures various interpretations and evaluations of the paper and interactions by these four learners.

1.1.2 Discussion

The above example illustrates a typical usage scenario of CLARE: it highlights a collaborative learning process that is guided by a well-defined representation language and a process protocol, and supported by a computer-based environment. Unlike traditional learning which takes place in laboratories or classrooms, the current process is supported by a virtual environment. This implies that Mary and Scott, for example, might be geographically and temporally distributed, but can still compare their interpretations of the selected paper and discuss their differences and similarities using CLARE. Perhaps more importantly, the example scenario represents a new type of learning called *collaborative learning from scientific text*. Figure 1.5 shows major components of this learning and they are related together to support collaborative knowledge construction. The boxes that are connected to the outmost circles indicate that learning in this context begins with scientific text (e.g., [KTBB92]), as opposed to scientific experiments or lectures. The process consists four steps: *summarization & evaluation*, *comparison*, *argumentation*, and *integration*. The direction of the large shaded arrows indicates that, as the process moves toward the center, the amount of interactions among the group members increases and, concurrently, the group knowledge converges. The ultimate result is a dynamic group knowledge base, which integrates various interpretations, evaluations, deliberations, and extensions of the subject content of the selected artifacts by a group of learners.

The outermost layer represents *summarization & evaluation*, corresponding to the CLARE's *exploration* phase. Its primary purpose is bootstrapping — to reconstruct the thematic structure of the knowledge embedded in the selected artifact. Summarization is very similar to reverse engineering in software development, which attempts to recover the design information embedded in the software source code. Evaluation, on the other hand, brings the learners' perspectives to bear with the learning context by allowing them to explicitly state their views on the content of the artifact. As shown in Figure 1.5 by the isolated clusters of nodes around each learner, the exploration phase is private; each learner

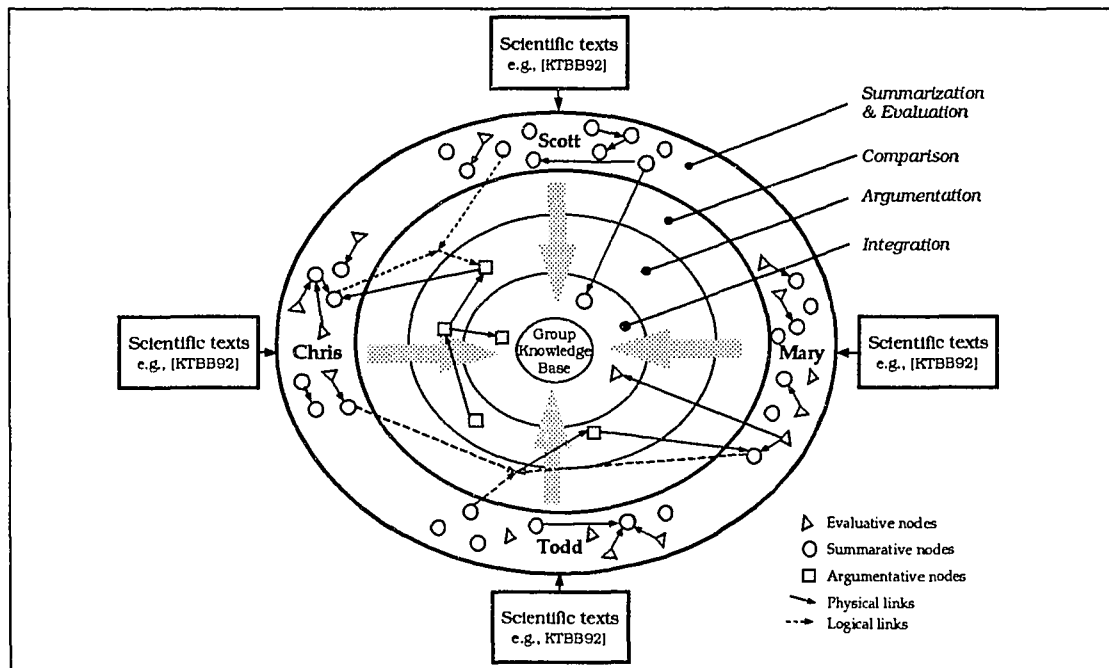


Figure 1.5: Collaborative knowledge-building in CLARE

independently derives his own representation and assessment. As a result, a learner cannot either be swayed by, nor free-ride off the work of others. The result from this step — the summarative and evaluative representations by individual learners — forms a basis for the second phase.

Because of the difference in backgrounds, interests, and intellectual perspectives of the learners involved, the representations from the previous step are likely to be different, as evidenced from the example. The *comparison* mode provides a convenient means of uncovering differences in the interpretation and evaluation, and ambiguities in the presentation. This mode forms the baseline for subsequent two steps — *argumentation* and *integration* — in which learners deliberate, extend, and integrate their interpretative and evaluative knowledge.

Collaborative knowledge-building in CLARE bears many similarities with knowledge-building in the scientific community (see Section 2.2). On this ground, CLARE deviates from many other existing learning systems in which collaborative learning is largely man-

ifested as information sharing (see Section 7.3 for a review of such systems). The next section explains the main motivation behind the current approach.

1.2 Motivation

This research is motivated by two main trends: one is technological and the other is theoretical. The former is the predominant emphasis on *access* by existing collaborative learning systems. The latter is the increasing recognition of the importance of meta-knowledge in human learning. While the technological force propels the need for computational support beyond information sharing, the theoretical development forms a conceptual underpinning for the current approach.

1.2.1 Technological biases in current learning support systems

The two most widely used types of collaborative learning environments are *virtual classrooms* and *hypermedia systems*. The former encompasses a wide range of computer-mediated communication technologies, including electronic mail, computer conferencing, and bulletin-board systems. The latter promises to deliver integrated learning environments that link together a wide range of applications and distributed data, such as text, graphics, video, etc. Despite their seemingly differences, virtual classroom and hypermedia systems share essentially the same focus, namely, support for information sharing. They both aim at overcoming the geographical, temporal, and media constraints of traditional face-to-face interactions and printed media by allowing the learner access to the *right information* in the *right media* or *presentational format*, or access to the *right people* at the *right time*. For example, computer-mediated communication systems, such as EIES [Hil88], have been successfully used to overcome the *same-time, same-place* constraints of traditional classrooms, and to increase student participation outside physical classrooms. Similarly, hypermedia systems, such as Intermedia [YHMD88] and NoteCards [HMT87], have been found effective for browsing and navigating large shared information space. However, both virtual classroom and hypermedia systems suffer from some major problems: *information overload* in virtual classrooms and *lost-in-the-hyperspace* in hypermedia environments.

At a deeper level, these problems are rooted in the same cause: the lack of explicit, fine-grained characterization and representation of the thematic structures of learning artifacts. In virtual classrooms, for example, online discussions typically take place within various *interest groups*. Such divisions are generally coarse-grained. The relationships between these groups are also *implied* rather than *explicitly* specified. In hypermedia systems, the power of linking and the emphasis on non-linearity often lead to the excessive use of such features. The net result, similar to the over-use of the *goto* statement in computer programs, is a network of nodes with spaghetti-like structures whose semantics are difficult, if not impossible, to understand. As a result, the potential for *deep-level* collaboration among a group of learners is severely limited.

At a more fundamental level, the above problems with virtual classroom and hypermedia systems are the manifestation of a techno-centric approach to learning support embedded in these systems. They indicate that insufficient attention has been paid to the underlying theories of human learning. The next section presents one of such theories — *assimilation theory of cognitive learning* — and its technological implications.

1.2.2 Cognitive learning theory and concept mapping

While technologists continue to improve the functionality and the interface of software and hardware tools, theorists are breaking new grounds in understanding human learning. One of the major developments in educational psychology is the theory of *meaningful learning*, also known as the *assimilation theory of cognitive learning*. The main premise of this theory is that the most important factor influencing human learning is the learner's prior knowledge; that human learning is evidenced by a change in the meaning of experience (as opposed to a change in behavior); and that the key role of the educator is to help students reflect on their experience and to continuously construct new meanings [Aus63, ANH78]. To facilitate this process, Novak and Gowin [NG84] — two of the main proponents of this theory — have developed two meta-cognitive strategies: *concept maps* and *Vee diagrams*. Both are the tools for representing changes in the knowledge structure of students over time, and for helping them *learn how to learn* (see Section 7.2.2 for details on concept maps and Vee diagrams).

Concept mapping represents the first true attempt to provide explicit support for meta-learning [NG84]. It has enjoyed a wide acceptance in the educational community. Numerous studies have shown its effectiveness in facilitating student learning in science [Cli90, Nov90, RR92]. Despite the strong empirical evidence supporting its usefulness, however, concept mapping as a collaborative learning tool suffers from three main problems:

- *Non-hierarchical structures:* In concept maps, all knowledge features must ultimately be reduced to concepts and links between them. This, though achievable for introductory, well-understood knowledge, is inadequate in advanced learning context, which often requires analyses and syntheses to be done using higher-level constructs, e.g., *claim, problem*. The lack of abstraction capabilities severely limits the usefulness of concept mapping for advanced learners such as graduate students.
- *Free form of expression:* Concept maps, like the designer's sketch pad, give the learner full freedom in deciding what to draw and how to draw it. The representation does not dictate nor provide any structural heuristics on how it should be used. While this flexibility makes concept maps extremely expressive, it also adds little structure on which useful manipulations can be applied, and which human learners may use to help them decipher the map. The latter is especially significant in collaborative settings, for this arbitrariness implies that it is difficult to compare, contrast, and integrate concept maps generated by different learners.
- *Individual learning tool:* Concept mapping has hitherto primarily been used in facilitating individual learning. Few existing systems support collaborative construction of concept maps. In their study of concept mapping in a group setting, for example, [RR92] have to rely on movable paper clips instead of a computer-supported environment.

The above problems with concept mapping indicate the need for extending the current strategy and for designing alternative strategies to support human learning. Moreover, the general lack of technological support for concept mapping has also prevented the realization of full potentials of such approaches. The current research is intended to address both of these problems.

1.2.3 Toward a theory-based collaborative learning support environment

This dissertation attempts to bridge the gap between the recent development in the theories of human learning and the current state of technological support for such activities. It does so by adopting the assimilation theory of cognitive learning as its conceptual basis, and by providing a new type of computer-based learning environment that focuses on collaborative learning as knowledge-building. It addresses the above problems with concept-mapping — the theorists' solution to the problem of facilitating meaningful learning — by proposing a new representation and a computational environment that supports the use and manipulation of this representation. The next section outlines the main thesis underlying this research.

1.3 Research thesis

The basic premise of the current research is that collaborative learning is not simply sharing of information among learners but rather collaborative knowledge construction similar to that taking place among researchers in the scientific community. One important form of collaborative learning is organized on scientific text, which attempts to bridge the gap between the knowledge-building in the scientific community and the knowledge-building in classrooms by systematically analyzing and discussing research literature. Scientific text is an important source of the most current, evolving knowledge. More importantly, it is one of the primary sources for learning about scientific knowledge-building itself⁶. The structure of scientific text often reveals patterns underlying scientific discourse and the norms governing formal presentation of research findings.

The central claim of this research is that CLARE provides a viable computer-augmented environment for collaborative learning from scientific text. First, CLARE does not follow the technology-driven paradigm. Rather, it is grounded in the assimilation theory of cognitive learning — a well-established learning theory, and constructionism — a

⁶Another way of learning the scientific knowledge-building is through direct participation, e.g., apprenticeship. The two are complementary rather than mutually exclusive.

widely adopted pedagogy. Second, CLARE incorporates a unique learning model that integrates such key activities as summarization, evaluation, comparison, argumentation, and integration. As illustrated in Figure 1.5, these activities form the basic building blocks of an artifact-based knowledge construction process. Third, CLARE is built on a thematically-based representation language which (1) draws on the basic principles of knowledge representation from AI, and (2) overcomes the three drawbacks of concept mapping — one of the main meta-cognitive tools proposed as part of the cognitive learning theory.

One of the main characteristics of this current research is its emphasis on the higher-level structure of knowledge or *meta-knowledge*, and the use of such knowledge as the *glue* that links together:

- Knowledge-building in the scientific community and knowledge-building in classroom settings;
- Exploration and consolidation phases in CLARE; and
- Different interpretations and points of view from individual learners.

The proposed representation language exemplifies what such high-order knowledge structure is; it specifies what content-level themes that the learner should attend to but does not dictate how such themes are to be found or used. It should be noted that the process of identifying such high-order constructs is an important form of meta-learning which CLARE aims to explicitly support.

1.4 Research contributions

This research addresses an important and yet hitherto untapped area of research — the explicit representation and the use of high-order knowledge as a means to facilitate collaborative learning. It addresses a broader issue of knowledge representation in the context of human learning. Despite that knowledge representation is at the central stage of AI research, the requirements for supporting human learning, especially, human collaborative learning, are quite different from that for AI systems. Along this direction, the current research raises a number of significant questions. For example, what is the exact role of the

representation in human collaborative learning? What are characteristics of the representation language appropriate for human learning? What types of computational augmentation are required in such a context? Though this research may not lead to definite answers to these and many other related questions, it does represent the first step toward the ultimate understanding of such important issues.

Conceptually, CLARE demonstrates a computer-supported environment that is based on constructionist pedagogy and the assimilation theory of cognitive learning. Despite that, pedagogically, the view of learning as knowledge-building is well-established, the technological support thus far has not yet extended beyond information sharing. This research highlights the connections between knowledge-building in the scientific community and knowledge-building in classroom settings by defining a new type of learning, called *collaborative learning from scientific text*. This learning is centered on the knowledge — both content and meta-level — embedded in scientific text. CLARE provides explicit process and representation support for such learning.

Representationally, RESRA — the CLARE's underlying representation language — applies the principles of knowledge representation to the thematic features of scientific text. It overcomes the structural weakness of concept maps by providing a small initial set of primitive types and canonical forms, which also serve as useful basis for exploring higher-order structure of human knowledge. The key contribution of RESRA is its potentials in integrating different points of view held by individual learners.

Technically, CLARE is both a research and a learning tool. As such, it provides an extendible computational environment that balances usability and the research-level support. The former is highlighted by the five-step process model that helps structure various learning activities, and by the comparison mode, which enables learners to make fine-grained comparisons of their own representations with those of others. The latter is evidenced by the built-in instrumentation mechanism that allows fine-grained process data to be gathered unobtrusively.

Empirically, the data from sixteen usage sessions of CLARE by six different groups of students from two separate classes confirmed that CLARE is a viable approach to supporting collaborative learning from scientific text. The result also suggests a number of

interesting directions in which CLARE can be extended and further empirical investigations can be conducted.

1.5 Scope and limitations

Collaborative learning is a complex activity to study and support. This research does not attempt to address all important aspects of the subject, nor create an all-purpose system that provides merely a collection of neat features. Instead, it focuses on the role of representation in collaborative learning, and on providing services which augment the human use and manipulation of this representation. Below are three major limitations of the current research:

- CLARE helps ameliorate certain problems related to face-to-face collaboration (e.g., the *dominant personality* effect) to the extent that it is a computer-mediated environment. However, it does not attempt to overcome the interpersonal and inter-cultural conflicts inherent in typical group settings. When used in a face-to-face setting, it is possible that the effectiveness of CLARE as an augmented learning tool be overshadowed by some interpersonal or inter-cultural factors. CLARE provides no means to separate the two.
- At the system level, CLARE does not have certain advanced functionalities found in other learning support environments. Examples include multi-media, version control, and fancy graphical interface. This is in part due to the pilot nature of the current implementation; new features will be added as the first-hand experience with the system increases and, consequently, the underlying process is better understood.
- Despite the importance of allowing learners to define their own representation primitives, this capability is currently unavailable to the user.

1.6 Organization of this dissertation

The remainder of this dissertation is organized as follows. Chapter 2 describes the conceptual framework for the current research. First, it redefines *knowledge representation* in

the context of human learning, and relates this concept to the theory of *meaningful learning*. Next, it discusses the role of scientific text in the knowledge-building process in both research and classroom settings. The following two sections introduce the representation language (RESRA), and the SECAI model of collaborative learning. The chapter concludes with a discussion about the role of representation in the proposed framework.

Chapter 3 describes RESRA — the proposed representation language based on the thematic structure of scientific text. It begins by identifying four major requirements for the representation. Then, it defines several key concepts underlying RESRA. The following two sections elaborate the two primary constructs of the representation: *primitive* and *canonical forms (CRFs)*. Examples are given to illustrate their semantics and usages. The final section of the chapter briefly discusses the extendibility of RESRA.

Chapter 4 presents the design and implementation of CLARE. It begins with the three types of requirements for the system: *conceptual*, *data*, and *usability*. Next, it discusses two main design considerations of CLARE: layered + object-oriented design and the decision on services over interface. The system architecture is described next, which is followed by the depiction of the interface features. A road-map of the system functionality is also provided. The chapter concludes with a brief history and status report.

Chapter 5 describes the experiments designed for evaluating CLARE. It first revisits research problem. Then, it describes the ten hypotheses that guide the evaluation experiments. Next, it identifies and relates the three types of empirical data for each hypothesis: *outcome*, *process*, and *assessment*, the procedures for collecting such data, and methods for analyzing them. Finally, it describes the two sets of experiments to be conducted, including the task, subjects, procedures, and the execution plan.

Chapter 6 presents findings on CLARE from the evaluation experiments. It begins with an overview of the experiment and its findings. The actual result presentation is organized into three parts. Section 6.3 describes the findings with respect to each of the ten hypotheses identified in Chapter 5. Section 6.4 discusses main issues that arose from the use of the RESRA representation, including a list of common representation errors abstracted from the usage data. Section 6.5 identifies several usage strategies employed by the learners during summarization. Section 6.6 presents a detailed analysis of one group session using CLARE. The purpose of this last section is to bring together all previous discussions with

a single case example, and to compare this example with the hypothetical usage scenario described in Chapter 1. This chapter concludes with a discussion on the major results from the CLARE evaluation.

Chapter 7 reviews prior work pertinent to the current research. It is organized into four sections. Section 7.1 reviews the theoretical work on which CLARE is based. More specifically, it covers constructionism and the assimilation theory of cognitive learning. Section 7.2 describes schema theory and related knowledge representation languages. Section 7.3 surveys major existing collaborative learning systems and empirical findings on them. This chapter concludes with a summary of the relationships between CLARE and the work being reviewed.

Chapter 8 concludes this dissertation. It begins with a review of the basic problem this research has attempted to address and the approach it has adopted. Next, it describes the major contributions this research has made in furthering the understanding of collaborative learning in computer-augmented environments. The final section of the chapter discusses several promising directions in which the current research might be extended.

Chapter 2

Toward a Representation-Based Approach to Collaborative Learning from Scientific Text

The term *collaborative learning* has many connotations, ranging from peer-tutoring to computer conferencing. This research concerns a specific type of collaborative learning, called *learning from scientific text*, which is centered on scientific literature, such as research papers, journal articles, monographs, and so on. The approach focuses on the thematic structure of scientific discourse embodied in the written artifact. More specifically, collaborative learning in this context involves the following key activities:

- To summarize the content of a scientific artifact by identifying and representing its key thematic features and the relationships between these features;
- To evaluate the content of an artifact through making critiques, raising questions, and suggesting improvements;
- To compare individual summarizations and evaluations to uncover ambiguities, inconsistencies, and differences and similarities;
- To clarify and resolve those ambiguities, inconsistencies, and differences through constructive argumentation; and
- To integrate similar and related points of view to form a coherent corpse of shared knowledge.

This chapter describes the conceptual basis of the above approach. It draws from several streams of research: knowledge representation in AI, cognitive learning theory, con-

structionism, and structural analysis of scientific text. In doing so, it formulates a theoretical framework for the entire project.

The chapter begins by relating the AI concept of *knowledge representation* to human learning. It points out how knowledge representation may be viewed as a means of understanding the high-order structure of knowledge. Second, it discusses the characteristics of scientific text, followed by an overview of RESRA — the new representation language for characterizing the thematic features of scientific text and for serving as a shared framework for collaborative learning. The chapter concludes by describing the representation-based model of collaborative learning from scientific text, and shows how it integrates knowledge representation, scientific text, and human learning into a single learning support environment.

2.1 From semantic nets to concept maps: knowledge representation in human learning

Knowledge representation (KR) is a central concern of artificial intelligence (AI). In essence, it is the core of all intelligent systems, including machine learning, intelligent tutoring, and expert systems. However, the concept of KR is rarely discussed with respect to human learning, especially collaborative learning. The purpose of this section is to examine human learning in terms of knowledge representation. In doing so, it highlights several important differences and similarities of KR in these two contexts. The emphasis is on the representation language instead of the information processing details by either human or machine. Since KR concerns the deep-structure of human knowledge, the view of human learning in terms of KR highlights the importance of meta-knowledge. In addition, it also forms a basis for the proposed representation-based approach to collaborative learning among human learners.

2.1.1 Two types of knowledge representation

In AI, the term *knowledge representation* (KR) is used to refer to the process of encoding various types of knowledge into a form with which a computer program can reason;

the embodiment of such knowledge enables the computer to perform certain tasks that normally require human intelligence. At the center of this transformation lies the representation formalism, ranging from formal logic and production rules to semantic networks and frames. These schemes define what knowledge to represent and how it is represented and manipulated inside the computer. When viewed from the context of human learning, however, knowledge representation denotes something quite different : it concerns *meaning extraction* from external knowledge sources, which include both written artifacts and living sources, e.g., researchers, teachers, peers. At the center of this process also lies the representation scheme, which determines what and how human knowledge is represented. The default human representation scheme is the natural languages, e.g., English or Chinese. There are also other special-purpose representations proposed to overcome certain deficiencies of the natural language. Concept maps, which will be introduced below, is an example KR scheme that is intended to facilitate human learning.

Figure 2.1 depicts knowledge representation in AI and human learning settings¹. At the process-level, the two possess a number of similarities. First, KR in both contexts consists of the same components: *knowledge source*, *agent* and *knowledge engineer* or *teacher*. The two also share the same goals: both aim at improving the level of knowledge and the ability to perform selected tasks, although one concerns a computational agent, while the other concerns a human learner. Third, they both acquire knowledge from the same sources, i.e., either codified, written artifacts such as research papers, or living sources such as experts in a given field², or both. Fourth, knowledge acquisition is accomplished using a selected KR scheme, and facilitated by the knowledge engineer or the teacher.

Despite these similarities, KR schemes used by human and computational agents differ in some fundamental ways. The dividing line between the two is the distinct roles of the computational agent and human learner. A typical AI program is merely a passive recipient of knowledge. In contrast, human learners are *actors* who *interpret* incoming information in light of their prior knowledge and experience, and give it context-specific

¹The analogy between the computer and the human mind has profound philosophical implications which go far beyond the scope of the current discussion. For an example discussion, see [Har89].

²The focus of the current research is on the codified, public knowledge, as recorded in the written artifacts, in particular, *primary* artifacts, such as research reports. See Section 2.2 for details.

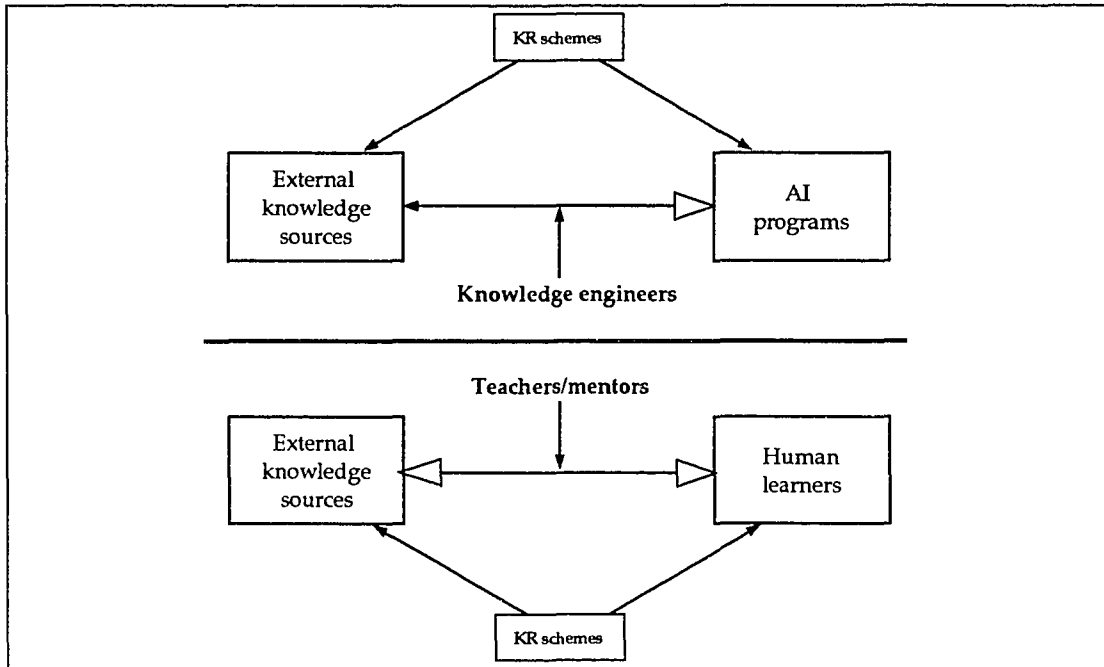


Figure 2.1: Knowledge representation in human learning and AI contexts

meanings. In other words, it is the learner who plays a decisive role in determining what and how knowledge is acquired. The teacher, on the other hand, serves mostly a facilitating role in the knowledge acquisition process. In an AI system, the knowledge engineer is instrumental in deciding what knowledge gets transferred to the target system. This role difference is shown in Figure 2.1 by the presence of the link with big arrows on both ends connecting human learner and the knowledge source. The small arrow pointing from the AI program to the knowledge source indicates that few such systems actually contribute to the public knowledge source, though some machine learning systems can improve their performance automatically by incorporating new knowledge from external sources.

At the representation level, the KR schemes used by computational agents and human learners differ in a number of ways. The specific differences between them are summarized in Table 2.1. In general, the representation used by the computer program is formal, that is, its syntax and semantics are precisely defined and enforced (e.g., semantic nets); restrictive in terms of what can be expressed and how it can be expressed; fine-grained, e.g., phrase or sentence levels as opposed to paragraph or artifact levels; and finally, a good AI represen-

tation is parsimonious and computationally efficient. In contrast, the KR scheme used by humans, as exemplified in the natural language, is informal, expressive, emphasizing on coarse-grained structures, likely to be redundant, and computationally inefficient and, in many cases, computationally intractable. A good example of the latter is natural language understanding.

Table 2.1: KR schemes used by human learners and AI programs

Criteria	AI Systems	Human Learning
Formality	Formal	Informal
Expressiveness	Restrictive in scope	Expressive
Granularity	Fine-grained	Coarse-grained
Parsimony	Concise	Redundant
Computational efficiency	Efficient	Inefficient

The above discussion of KR and the comparison of KR requirements for human learning and machine reasoning is relevant to the current context in two significant ways. First, representation is important not only in AI systems but also in human contexts, especially in ill-structured domains such as learning and design, and in information-overloaded environments such as hypermedia and virtual classrooms. AI researchers have developed a large number of techniques and representation schemes which can be readily applied to other contexts, including human learning. In fact, this approach has already been applied to the domains such as design rationale management [LL91, CY91]. Second, KR is either ontological or epistemological [Swa90]. In other words, KR concerns the higher-order structure of human knowledge, or meta-knowledge. Many KR schemes, such as frames, semantics nets, Schank's conceptual dependency theory, are themselves examples of meta-structures. As described in the following section, the importance of higher-order knowledge in human learning is increasingly recognized by educational researchers. To facilitate the use of such knowledge, they have in fact proposed their version of KR schemes, which

they call *meta-cognitive tools*. Concept mapping, to be described in Section 2.1.4, is such an example.

2.1.2 Content learning, meta-learning, and knowledge representation

Human learning can be viewed at two levels: *specific* and *generic*. The former involves the understanding of a specific subject matter; hence, it is also referred to as *content learning*. The latter, which is commonly called *meta-learning*, concerns the nature and the structure of what is being learned, as well as the process through which knowledge is acquired. For example, at a content level, when a student is first exposed a programming language such as C, he learns the syntax and semantics of that language. At a meta-level, he might attempt to draw an analogy between a programming language and the natural language, or be interested in general strategies in acquiring language skills. More specific distinctions can be made between the two at four different levels: *semantic orientation*, *relational knowledge*, *process knowledge*, and *learning strategies*, as summarized in Table 2.2. In general, content learning emphasizes the direct meanings from the snapshot of isolated learning artifacts. Meta-learning, on the hand, views each artifact as an episodic component of the overall knowledge of a given field. Its focus, therefore, is on the deep structure and semantics, as well as the inter-connections among various knowledge chunks. In addition, it also attaches a central importance to knowledge acquisition, i.e., how learners make sense of artifacts presented to them. In terms of learning strategies, meta-learning is often associated with *meaningful learning* (to be described below), while content learning is often realized through *rote learning*. In typical classroom settings, content and meta-level learnings are often intertwined. At a content level, for example, participants in a research seminar are expected to understand the particular subject matter that is under concern (e.g., AI). At a meta-level, they need to learn how to collaborate, how to research literature, how

to present and evaluate research artifacts, how to identify interesting problems and develop novel solutions, and so forth.

Table 2.2: Characteristics of content and meta-learning

Category	Content learning	Meta-learning
Semantic orientation	Surface, direct meanings	Deep, embedded meanings
Relational knowledge	Isolated pieces of knowledge	Inter-connections among various knowledge chunks
Process knowledge	Static, final version of knowledge	Process of knowledge acquisition and changes of meanings over time
Learning strategies	Mostly, rote learning	Meaningful learning

The distinction between content learning and meta-learning is significant for three reasons. First, although content learning tools exist and are improving, there are few tools to support meta-learning. For example, concept mapping is still mostly done manually. Second, meta-learning has become increasingly important because of the accelerating rate of knowledge production and dissemination, students may find the subject content they learn in school quickly becoming obsolete, but any meta-knowledge and skills they acquire will enable them to better adapt and cope with the changing state of human knowledge. Third, the differentiation between content and meta-learning clarifies the role of knowledge representation in human learning. As shown in Figure 2.2, the links from knowledge representation to meta-learning, and from meta-learning to content learning are *direct* ones, as expressed the solid line, while the link from knowledge representation to content learning is an *indirect* one, as shown in a dashed line. For example, at a content level, when a student from a software engineering class is asked to read a research paper on that subject, he may simply browse through it and learn nothing. However, he can probably get a much deeper understanding of the paper content by using a representation such as RESRA. On the other

hand, if he is asked to merely study RESRA independent from his learning context, he may not find RESRA helpful or relevant.

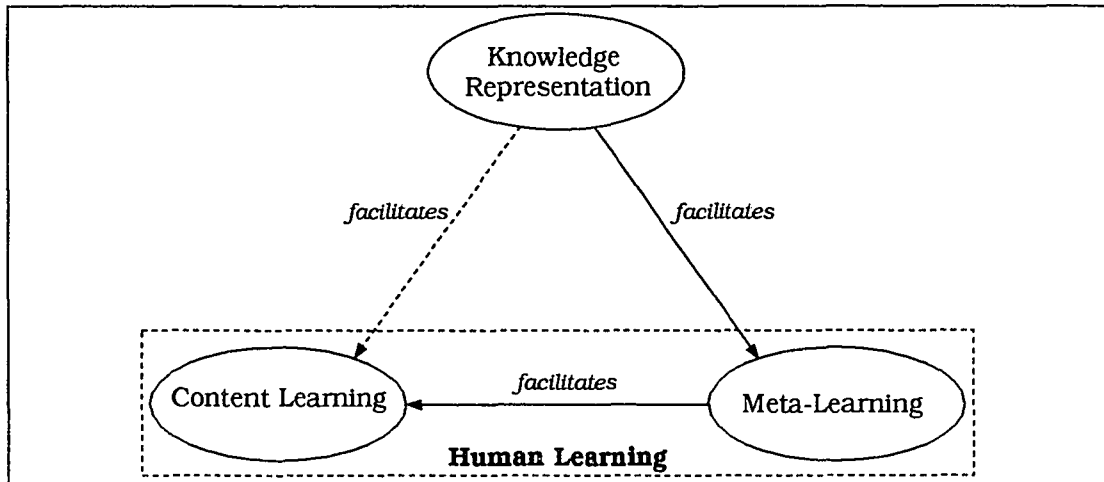


Figure 2.2: Relationships between knowledge representation, meta-learning, and content learning

2.1.3 Cognitive learning theory and concept mapping

One major development in educational psychology is the *theory of meaningful learning*, also known as the *assimilation theory of cognitive learning* or simply, theory of cognitive learning. It has evolved at Cornell University over the past three decades [Aus63, NG84]. The thrust of the theory is its emphasis on the importance of meta-learning, that is, *learning how to learn*, and the role of the meta-knowledge in human learning. The basic tenets of this theory include:

1. The single most important factor influencing human learning is what the learner already knows, i.e., prior knowledge;

2. Learning is evidenced by a change in the meaning of experience rather than a change in behavior, in contrast to the view long held by behavioral psychologists; and
3. The key role of the educator is to help students reflect on their experience (and hence give it new meanings), and construct meanings from the artifact in light of the changing experience.

The view that learning is *meaning-making* places knowledge representation at the focal point of the human learning process. KR defines not only the form in which certain type of knowledge is highlighted to the learner, but also the process by which such a form is derived. In addition, KR schemes, which are called *meta-cognitive tools* by educational researchers, are the standard language for characterizing both knowledge structures and corresponding cognitive structures. They help the learner differentiate and organize newly acquired meanings. As part of the theoretic formulation, Novak and Gowin [NG84] propose two knowledge representation schemes: *concept maps* and *Vee diagrams*. Figure 2.3 shows an example of the concept map. The next section describes concept mapping — a scheme that has been found quite effective in enhancing science teaching [Cli90, Nov90, RR92] — from a KR perspective, and identifies the problems that concept mapping shares with semantic networks.

2.1.4 Critiques on concept maps: a KR perspective

Concept maps are similar to semantic networks — a widely used KR scheme first proposed by [Qui67] — in that both represent knowledge as a network of inter-connected nodes, where nodes correspond to concepts, and links correspond to various types of relationships between these concepts. While semantic nets are constructed by trained knowledge engineers solely for computational manipulations, concept maps are built by learners themselves as a means of understanding knowledge. The two also differ in a more pro-

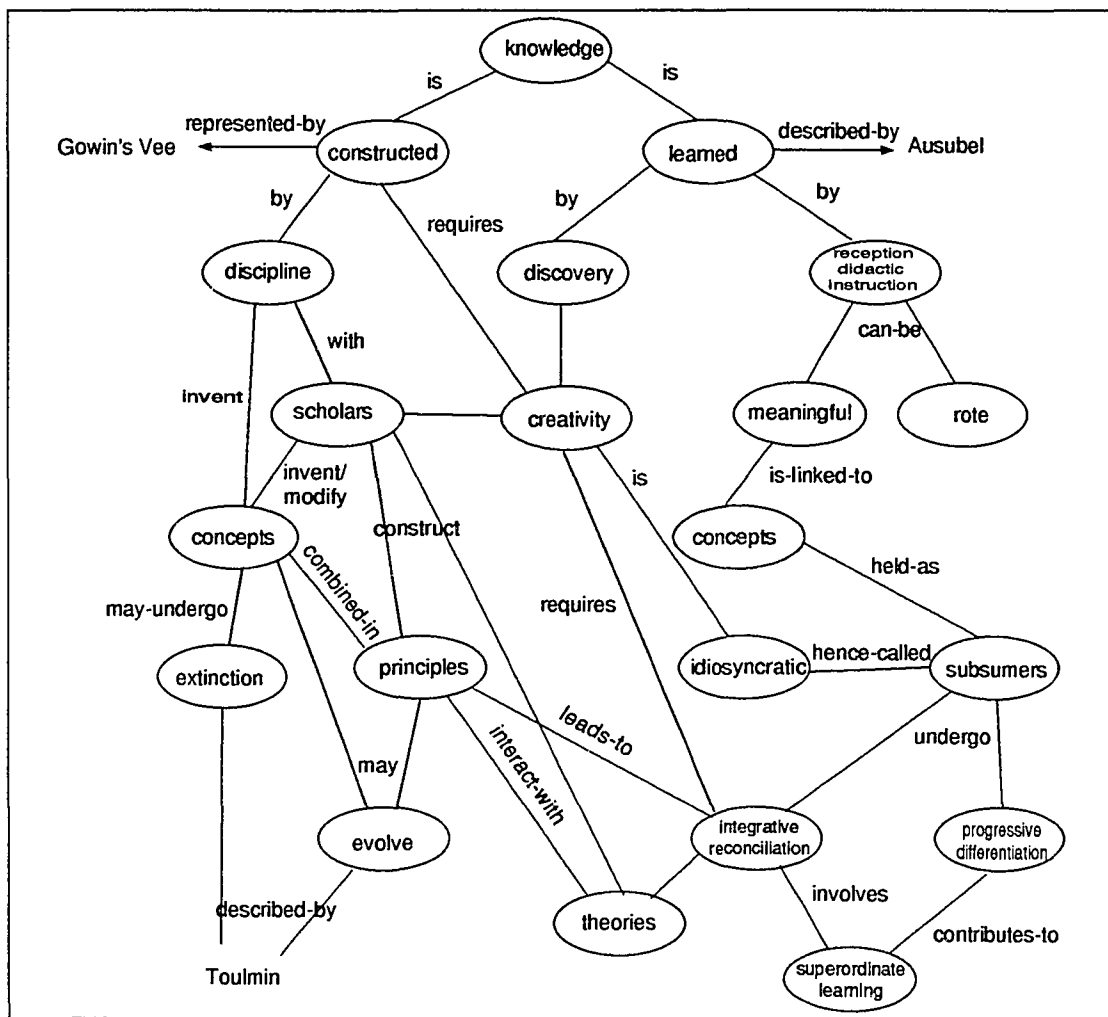


Figure 2.3: An example concept map on knowledge construction and acquisition from the perspective of the assimilation theory of cognitive learning (from [NG84])

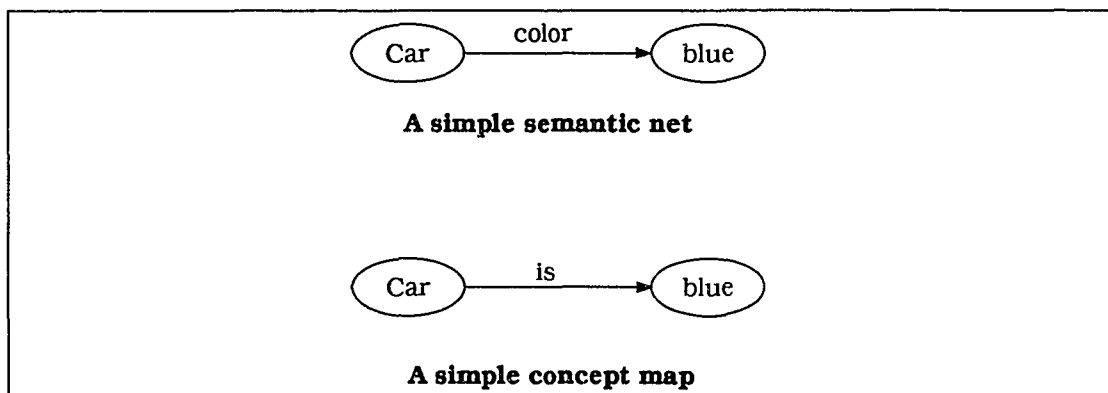


Figure 2.4: Concept maps versus semantic nets: an example

found way. Figure 2.4 illustrates a concept map and a semantic net representation of “the car is red.” Although the two express the same proposition, they use two very distinct link labels: the semantic net uses “color” to indicate that “red” is the value of the attribute *color* (instead of, for example, *make*); the concept map, on the other hand, uses the generic verb *is* to express the same semantics. The latter expression, albeit more readable, is also more ambiguous, which is a clear remnant of the natural language.

Despite the above differences, concept maps share certain problems with semantic nets. One such a criticism concerns their semantics: although different types of nodes and links are used in semantic networks, their exact semantics are often not specified or left ambiguous [Woo85]. The simple network in Figure 2.4, for example, may mean either the definition of the concept of a blue car, or the assertion that some car is blue. Like semantic nets, the concept map does not restrict the type of nodes and links to be used, and hence gives the learner complete freedom of deciding what to represent and how to represent it. This flexibility, while making concept maps extremely expressive, adds little structure that can be used as a basis of computation, and/or as an aid to human learners in making sense of the map. The latter is especially significant in a collaborative setting, for the lack

of shared semantics will invariably make it difficult to compare, contrast, and integrate concept maps generated by different learners.

The other major limitation of the concept map is *granularity*: all knowledge must ultimately be reduced to concepts and propositional links between them. Though such constraint is acceptable for introductory knowledge, it is inadequate for advanced learning (e.g., graduate students). The latter often requires analysis and synthesis to be done at a much higher levels, such as “claims,” “problems,” “questions,” “goals.”

The above deficiencies of concept maps call for an alternative scheme that addresses issues specific to collaborative learning from scientific text. The next section discusses the role and structure of scientific text. It also introduces RESRA, the successor to concept maps.

2.2 Representation of scientific text

Learning and research are traditionally viewed as very distinct activities: one concerns the production of knowledge and the other, the acquisition or reproduction of knowledge. This view, however, has been challenged by constructivism, a currently dominant paradigm in both the sociology of science and educational research [BL66, KC81]. From the constructionist point of view, learning is knowledge-building. Like science, learning involves such activities as problem identification, theorizing, hypothesis formulation and testing, refutation, and so on. Also like science which is primarily a social activity involving a community of scientists who share the same disciplinary knowledge, learning takes place in the context of a community of learners who interact with each other in an attempt to deepen their understanding of a particular subject domain.

This section is based on the premise that scientific text is not only a primary source of

human knowledge but also an embodiment of the norms governing scientific knowledge-building. It first discusses the role of scientific text in scientific knowledge-building and in human learning. Then, it describes three types of structures: *presentational*, *rhetorical*, and *thematic*. It concludes by introducing RESRA — the representation that is based on the thematic structure of scientific text, and intended to serve as a basis of collaborative learning.

2.2.1 The role of scientific text

The term *text* is used here in a broad sense to refer to any type of written record of human knowledge and experience, including letters, working papers, technical reports, journal articles, monographs, and so forth. It comes with any form of media, printed, audio, and video. Written text plays a vital role in scientific knowledge-building. Specifically, they serve the following four purposes:

- *As a formal channel of scientific communication.* Communication among researchers is done at two levels: *informal* and *formal*. The former includes direct interactions among scientists such as face-to-face conversations taking place in laboratories, hallways, conference rooms, or via telephones. The latter includes indirect exchanges through writings, e.g., letters, workshop and conference submissions, journal articles, monographs. The formality of the written text is based on the fact that it can exist, and thus be evaluated independent of the originator. The latter may lead to a higher-level of objectivity. The boundary between the two, however, is increasingly blurred by the wide use of electronic media. E-mail, for example, is used for both informal and formal purposes. Similarly, digital journals are gradually acquiring an equal level of formality as their printed counterparts [Har91, Gai92].

- *As a measure of professional accomplishments.* The quantity and quality of written publications are often considered as a key indicator of research productivity and, hence, the basis of promotion, recognition, and prestige within the scientific community.
- *As a primary repository of human knowledge.* Unlike textbooks, which contain pre-digested, often carefully-filtered knowledge, written text from the research front provides *knowledge-in-progress*, which may range from very preliminary ideas to coherent, well-established theories. They also allow the student to see how conflicts among competing scientific explanations are resolved, and how early explanations succeeded by more recent and, presumably, more valid ones.
- *As an important data source for studying scientific discourse.* Scientific text is seldom a verbatim record of what actually takes place in the laboratory or on the field. Rather, the decision on what to report (and not to report), and how to report it is often the result of a complex social process that involves the interplay of many factors. Scientific publications, like other types of writings, are rhetorical artifacts whose structures are shaped by the then-dominant paradigm of a particular field. Since it is not always possible to study scientists in their working place, scientific text is increasingly being used as a source to study the process of knowledge-building among scientists and the evolution of human knowledge over time [Sel93, Baz88]

From the learner's perspective, the differentiation of the above functions is significant in two ways. First, when learning in a classroom setting is viewed as knowledge-building, written text may be used for the same purpose as they are used in the real research context, for example, as a formal means of sharing knowledge among learners, and a primary data source for studying "classroom discourse." Second, the last two functions described above are especially relevant to the learning context. Written text from the research com-

munity contains “knowledge-in-progress” and embedded discourse structures of the scientific knowledge-building that are often absent from standard textbooks or lectures. By studying and analyzing them, the student can gain a better understanding of the nature of human knowledge as well as the process through which such knowledge is constructed and evolved.

2.2.2 The structure of scientific text

The structure of scientific text, that is, the ways in which scientific theories, findings, and evidence are presented in the form of written artifact, often varies from one discipline to another, and from one type of artifact to another (e.g., laboratory experiments versus field studies). Although it is not plausible to enumerate all possible structural types, scientific text can be analyzed at three levels: *presentational*, *rhetorical*, and *thematic*. At the presentational level, a research artifact is broken down into a hierarchy of chapters, sections, subsections, et al. Corresponding to each is a header, such as “abstract,” “introduction,” “experimental design,” “related work,” “conclusions.” Such structures provide useful pointers to the *type* of content that is immediately followed. However, they do not represent the content itself. Presentational structures are standardized through stylistic guidelines established for a given discipline or journal.

The second type of structure is *rhetorical*, which concerns the way in which scientific arguments are presented, supported, and refuted. Rhetorical models, such as the one by [TRJ84], provide a viable means of understanding competing formulations or explanations about a given phenomenon. In particular, they are useful for representing interrelationships among different artifacts, and the evolution of scientific formulations over time. For example, a journal might publish a special issue on a selected topic, that consists of a single “feature” article and a series of “reaction” articles written by researchers from different “schools of thought.” The structural pattern of these papers can be captured using

a rhetorical model. Interest in the rhetorical structure of scientific text has been mounting in recent years [Baz88, Sel93, Sim90]. Research efforts have also started in applying the same approach to human learning (e.g., [CSGMW92]).

The third type of structure is *thematic*. As the name implies, the thematic structure characterizes the important features or *themes* of scientific text, and the relationships between these themes. Two example *themes* are problem and claim, and an example relationship between them might be claim $\xrightarrow{\text{responds-to}}$ problem. Like the rhetorical structure, the thematic structure is content-oriented. Compared to the rhetorical structure, the thematic structure is more general and flexible: thematic models can be developed to capture both discursive and domain structures, both intra-artifact and inter-artifact relationships. For example, a thematic model might include primitives such as concept, claim, which, when instantiated into the domain of software engineering, can include “software complexity” (concept), “object-oriented-design” (concept), and “Object-oriented design offers a viable solution to software complexity” (claim). RESRA – the representation to be introduced next and described in details in Chapter 3 — is based on the thematic model.

2.2.3 RESRA: a KR scheme for representing scientific text

Given the importance of knowledge representation and scientific text in human learning, and the limitations of concept mapping (see Section 2.1.4), a new representation scheme is proposed. This new scheme, called RESRA³, is intended for modeling the thematic structure of scientific text, and for serving as a shared framework for collaborative learning.

RESRA provides 11 node and 20 link primitive types (see Table 2.3 for a summary

³RESRA, which stands for “REpresentational Schema of Research Artifacts,” is a specialized language for representing the thematic structure of research and learning artifacts generated from both within and without classrooms. A detailed description of RESRA constructs is provided in Chapter 3.

of RESRA node types, and Figure 3.1 and 3.2 for link types). The derivation of these primitives is based on several sources, both theoretic and empirical. The main theoretical sources are two: Vee heuristic [NG84] and Bloom's taxonomy [Blo56]. The empirical sources include case analyses of the structure of various artifacts and a number of rounds of experimental use of the representation.

The relationship between RESRA and concept mapping is similar to the relationship between semantic networks and Schank's conceptual dependency theory (CD) [Sch75]: CD responds to the problem of the lack of semantics in semantic networks by proposing a small set of node and link primitives that can be used to represent basic conceptual categories and the relationships between them. Similarly, RESRA overcomes the problem related to the unconstrained form of expression in concept mapping with its own set of node and link primitives. Furthermore, RESRA also addresses the fine-granularity problem of concept mapping: RESRA nodes are not limited to atomic constructs, such as "concepts"; they may be used to represent any type of complex propositions.

RESRA belongs to the content theory of knowledge representation; it concerns the type of knowledge that needs to be represented but not how this knowledge is used by human learners [Swa90]. Two major criticisms have been raised on the content theory: one is related to the large amount of efforts often required to translate, for example, a sentence into the underlying representation. Furthermore, the mapping from the text under study into the primitives is usually not unique. The latter, however, does not constitute a problem in RESRA since it is expected that individual learners construct different representations of the same artifact. The existence of these individual differences is a prerequisite for group synergy. The first problem (i.e., time-consuming), however, still exists in RESRA.

The second criticism is that primitives supplied by the content theory is usually incomplete with respect to the world it attempts to model [Swa90]. For example, it is likely that one may find thematic features that do not neatly fit in the existing node categories

Table 2.3: A synopsis of RESRA primitive node types

Node Type	Description	Example
Problem	A phenomenon, event, or process whose understanding requires further inquiry;	Meta-learning is not adequately supported by existing tools.
Claim	A position or proposition about a given problem situation.	Cleanroom engineering provides a viable solution in producing zero defect software.
Evidence	Data gathered for the purpose of supporting or refuting a given claim.	The use of cleanroom techniques has yielded a 10-fold reduction of defects in the project Alpha.
Theory	A systemic formulation about a particular problem domain, derivable through deductive or inductive procedures.	Ausubel's theory of meaningful learning.
Method	Procedures or techniques used for generating evidence for a particular claim.	the Delphi study, nominal grouping techniques, the Waterfall system development model.
Concept	A primitive construct used in formulating theory, claim, or method.	meta-learning; knowledge representation.
Thing	A natural or man-made object that is under study.	Atom, NoteCards.
Source	An identifiable written artifact, either artifact itself or the pointer to it, i.e., surrogate or reference.	An article by Ashton; the notes from Kyle's talk.
Critique	Critical remarks or comments about a particular claim, evidence, method, source, et al., or relationships between them.	The example applications of cleanroom engineering so far have been limited to well-defined domains.
Question	Aspects of a claim, theory, concept, etc., about which the learner is still in doubt.	How does box-structured design differ from object-oriented approach?
Suggestion	Ideas, recommendations, or feedbacks on how to improve an existing problem statement, claim, method, et al.	I would like to see cleanroom engineering used in some non-conventional domains, such as groupware.

of RESRA. However, RESRA, unlike CD, does not claim that the primitive set is exhaustive. In contrary, it recognizes that, in human learning, it is neither possible nor desirable to identify all potential primitive types at the outset. The undesirability is related to the usability of a representation language: the larger the set of primitives, the less usable the representation language, because the more efforts the learner has to devote to learn and use that language. RESRA allows the learner to define their own primitives if necessary.

One of the main characteristics of RESRA is that it is designed in conjunction with the SECAI learning process. RESRA primitives, for example, can be partitioned into *layers* corresponding to the steps in the SECAI process, e.g., summarization, evaluation. The next section defines a model of collaborative learning and highlights the importance of representation in that model.

2.3 Toward a representation-based model of collaborative learning

While RESRA provides a conceptual framework for representing the structural features of scientific text, the process-level question remains open: how do learners go about learning collaboratively from scientific text? This section describes SECAI, a two-phase, five-step process model of collaborative learning⁴.

2.3.1 Collaborative learning from scientific text

Although it may take place in individual, isolated contexts, learning from scientific text can be most profitably done in a collaborative setting. The reason is twofold. First, since

⁴SECAI stands for the five key components of the collaborative learning model, i.e., Summarization, Evaluation, Comparison, Argumentation, and Integration.

scientific text is undigested and hence less systematic compared to textbooks and lectures, interpreting and understanding its content is not always an easy task. Because of the differences among learners in their backgrounds, skills, perspectives, and experiences, they are likely to have distinct views on the artifact. A group setting permits the learners to share their views with each other, and thus enables all of them to gain a better understanding of the artifact. Second, like scientific research, learning is a process involving social construction of knowledge: learners should not only learn from scientific text but more importantly, from each other through perspective and knowledge sharing. In addition, since scientific text is the product of the collaboration among scientists, it sets an example for the learners on how to form their “knowledge-building” communities, how to jointly construct new knowledge among themselves, and so forth.

2.3.2 SECAI: the collaborative learning model

The SECAI model defines a process for collaboratively learning from scientific text. The process consists of two phases: *exploration* and *consolidation*. Exploration requires interpretations and evaluations of the content of a scientific artifact by individual learners. It in turn is composed of two steps: *summarization* and *evaluation*. Exploration is done privately; each learner must independently derive his own representation and assessment. As a result, a learner cannot either be swayed by, nor free-ride off another learner’s points of view. The result from this step, namely, the summarative and evaluative representations by individual learners, forms a basis for the subsequent consolidation phase.

The consolidation phase is the *public* phase of the SECAI process. It consists of three components: *comparison*, *argumentation*, and *integration*. Consolidation brings together individual interpretations and views on a given artifact. It allows learners to compare, question, critique, defend, relate, and integrate these interpretations and evaluations. As a

Table 2.4: Five principal activities of collaborative learning from scientific text

Activity	Description
Summarization	Extracting, condensing, and relating important themes from an artifact.
Evaluation	Subjective assessment of merits and soundness of the author's work.
Comparison	Finding out and highlighting differences and similarities among different points of view.
Argumentation	Challenging others' positions; defending one's own positions; proposing alternative solutions.
Integration	Declaring similarity, subsuming, same-perspective relationships between nodes; endorsing nodes and links.

result of these interactions, learners can develop a deeper and more complete understanding of the underlying artifact, and the perspectives of other learners. As a side benefit, they also create a group knowledge base that captures the above deliberation process.

The five process steps of SECAI, namely, *summarization*, *evaluation*, *comparison*, *argumentation*, and *integration*, are summarized in Table 2.4. Figure 2.5 shows the relationships between the five components. It illustrates SECAI as a process of collaborative knowledge-building based on the scientific artifact. The world outside the outmost circle contains scientific text that forms the starting point of learning. The big shaded arrows indicate the direction of the group process, which begins with *summarization & evaluation*⁵. It also shows that, as the process moves inward, the amount of interactions among the group members increases and, concomitantly, the group knowledge converges. The ultimate result is a dynamic group knowledge base that integrates different interpretations, deliberations, and extensions of the subject content of the artifact.

⁵There are two main reasons for showing summarization and evaluation as one combined step in this diagram: (1) the emphasis of this diagram is on the group collaboration, while both summarization and evaluation are individual activities as defined in SECAI; and (2) Both activities are viewed as bootstrapping in the current view, and they are often invoked in an intertwined rather than sequential manner.

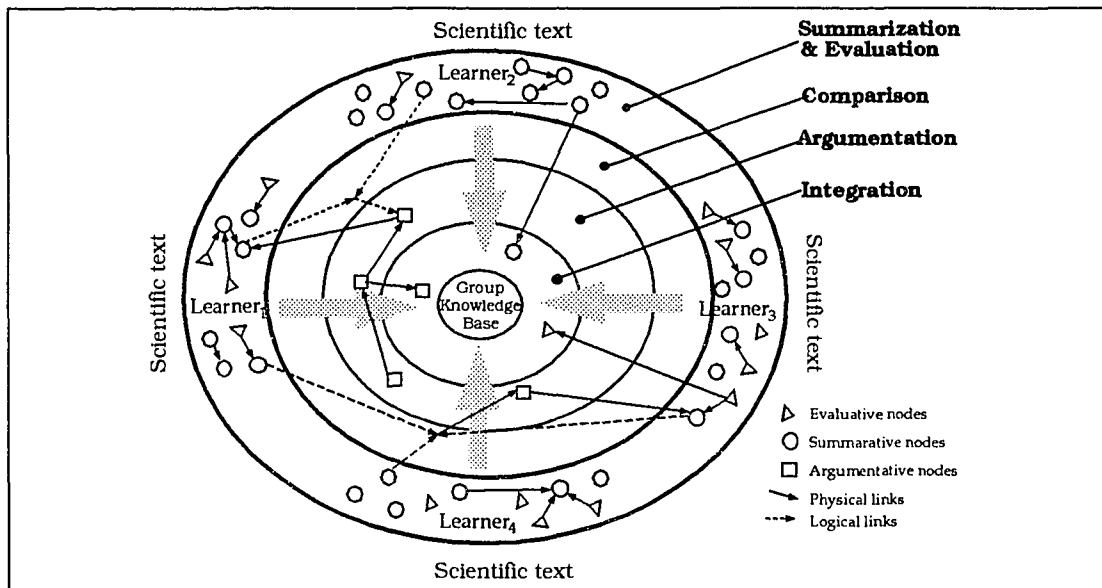


Figure 2.5: Collaborative learning from scientific text

2.3.3 The role of the RESRA representation

Figure 2.6 presents another view of SECAI. This view emphasizes an indispensable role of RESRA in the collaborative learning process. It shows that none of the SECAI activities is unbound, open-ended. Rather, they are all guided and constrained by RESRA, which serves as a *glue* that ties together:

- *Exploration* and *consolidation* phases; and
- Different interpretations and points of view from individual learners.

In addition, since the thematic features of scientific text are *summarized* and *evaluated* in terms of RESRA, and that the group deliberation is also done within the RESRA framework, the representation also forms a bridge that connects together knowledge-building in the scientific community and knowledge-building in CLARE-mediated classroom settings.

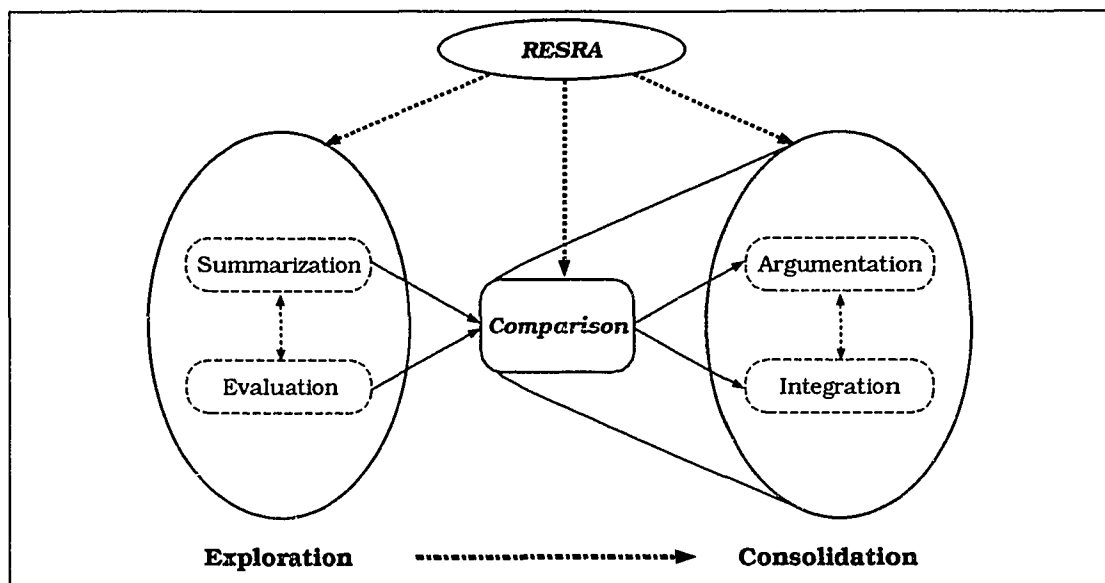


Figure 2.6: The role of RESRA in collaborative learning

2.4 Summary and conclusions

This chapter focuses on knowledge representation in human learning, in particular, human meta-learning. It attempts to integrate the technological and representational developments of KR into the conceptual framework of the cognitive learning theory to form a theory-based approach to supporting human learning. In doing so, it identifies several problems with concept mapping — the theorists' proposal to support meta-learning. The importance of scientific text in human learning is highlighted, and the structure of scientific text is described. A new representation language called RESRA is proposed to overcome certain weaknesses of existing KR schemes, and to provide unique support for collaborative learning from scientific text. This representation, along with the SECAI process model, forms the conceptual basis of the CLARE system. The next chapter describes the RESRA representation language, and Chapter 4 describes the CLARE system.

Chapter 3

RESRA: the Representation Language

The previous chapter discusses the motivation and the conceptual framework for a representation-based approach to collaborative learning. It also introduces RESRA — a special-purpose language designed for representing the thematic features of scientific text, and individual learners' points of view. This chapter describes RESRA in detail. It begins by identifying four major requirements for the language. Next, it defines several key concepts underlying RESRA. The subsequent two sections elaborate the two main constructs of the representation: *primitive* and *canonical forms*, respectively. Examples are given to illustrate their semantics. The chapter concludes by discussing the extendibility of RESRA.

3.1 Representational requirements

RESRA falls into the genre of the semi-structured representation. Unlike traditional knowledge representation schemes (e.g., predicate logic, frame) which aim at formalizing knowledge for machine reasoning, the primary purpose of the semi-structured representation is to aid human reasoning and communication. RESRA, in particular, is designed to help human learners extract meanings from research artifacts, and facilitate interactions among learners. Below are the main requirements for the language:

- *RESRA should represent the essential feature of a research artifact.* The term *feature* encompasses individual *thematic components* of an artifact as well as the relationships between those components. Although the focus of RESRA is on the *essential feature*, the *essentiality* of a thematic component is by no means absolute: a learner can conceivably take a minor point made by the author and treat it as something major, because it happens to be of interest to him. Hence, RESRA needs to be able to express a full range of thematic features.

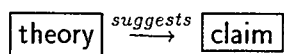
The key to the expressiveness of RESRA lies in the granularity of its primitives: if the grain size is too coarse, it sacrifices the expressiveness of the representation, i.e., less differentiating; on the other hand, if the grain size is too fine, the primitives are likely to be incomplete, and more difficult to learn and use. The design of RESRA needs to strike a delicate balance between the two.

- *RESRA is capable of representing various learners' views on the content of an artifact.* Similar to the previous requirement, the most important consideration here is the right grain-size, i.e., different levels of views, such as evaluative views, constructive views, et al.
- *RESRA mirrors the structure of the CLARE learning model.* Specifically, RESRA primitives should be divided into four groups: *summarization*, *evaluation*, *argumentation*, and *integration*; each supports the corresponding component in the SECAI learning model.
- *RESRA is usable by human users.* One important criterion of usability is *parsimony*, i.e., the initial set of primitives is small enough that their semantics can be mastered by human learners in a reasonable period of time. To support a large primitives set or to allow learners to create their own primitives, RESRA needs to provide mechanisms for domain specialization.

Though it is conceivable that RESRA be used in a paper-and-pencil mode, the representation presupposes a computer-based support environment. One main benefit such an environment provides is the capability to make the primitive selection context sensitive and, as a result, increases the ease of using the language.

3.2 Basic concepts

RESRA, or the REpresentational Schema of Research Artifacts, is a language that combines aspects of two widely used knowledge representation schemes: semantic networks and frames. At core, RESRA consists of two constructs: *primitives* and *canonical forms*. The former is the atomic building-block of RESRA. It is composed of two genres: *node* and *link*. A node is used to represent the essential thematic feature of a learning artifact and the learner's points of view. A link is for describing the inter-relationship between any two nodes. For example,



where theory and claim are node primitives; *suggests* is a link primitive; the expression as a whole is called a *tuple*.

A RESRA node typically consists of a number of *fields*; each describes one aspect of the targeted theme. For example, the node primitive `claim` consists of such fields as “name,” “type of claim,” “description,” and so on. RESRA links are often referred to as tuples, for example, `evidence` *suggests* `problem`. The process of creating an *instance* of a node or link primitive is called *instantiation*. Node and link primitives are described in the next section.

The *Canonical RESRA Form*, or CRF for short, is a collection of inter-related tuples that, together, describe an exemplary thematic structure of a particular type of artifact, such as a concept or empirical paper. While the primitive is useful for revealing the fine-grained components of an artifact, CRF is for highlighting the artifact-level features. The two are complementary. CRFs are described in detail in Section 3.4.

3.3 RESRA primitives

RESRA primitives are divided into four groups: *summarative*, *evaluative*, *argumentative*, and *integrative*, which mirror the four types of primary activities in the SECAI model. The following section describes each category in turn. A graphical depiction is also provided.

3.3.1 Summarative primitives

Summarative node and link primitives are intended for *summarizing* the content of a learning artifact, a process similar to reconstructing design information from program source code. CLARE provides 9 summarative node primitives and 14 link primitives, which are described below. Figure 3.1 is the corresponding graphical depiction.

Summarative node primitives

Problem: A phenomenon, event, or process whose behavior cannot be fully explained based on the current state of knowledge and, hence, requires further inquiry. An example **problem** is: “Despite rapid improvement of development environments and testing techniques, software systems still contain bugs.” Note that a **problem** is not a *question*,

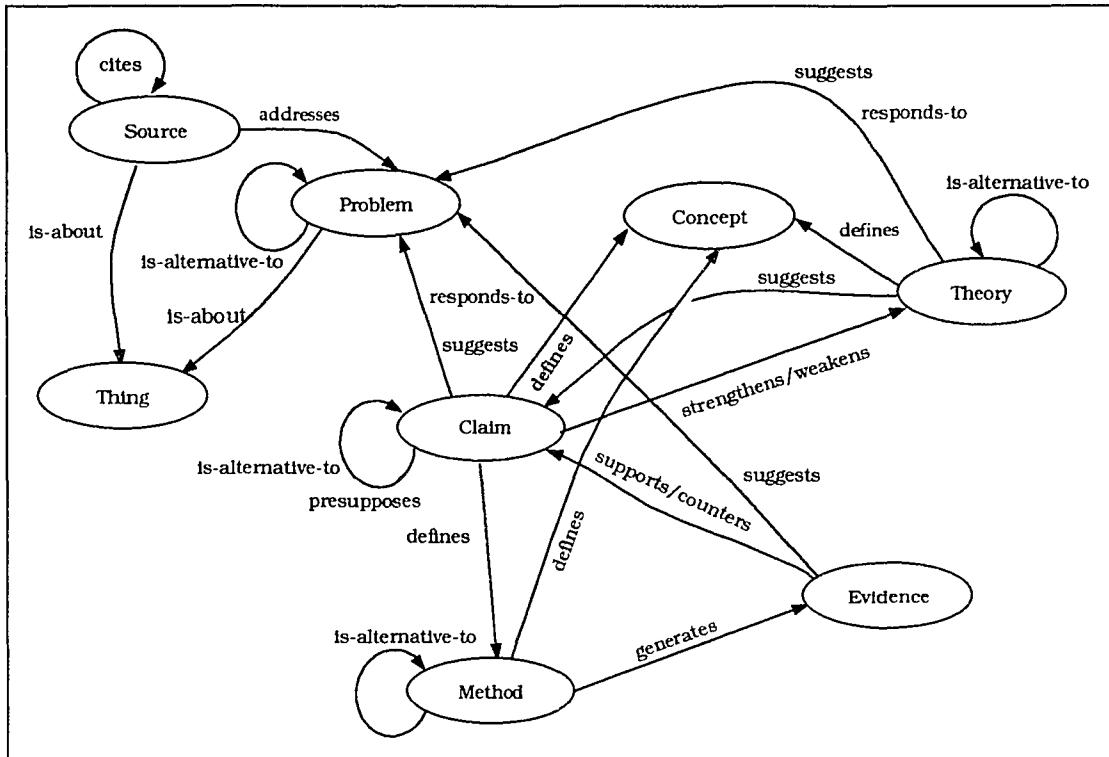


Figure 3.1: RESRA summarative primitives at a glance

though it is often triggered by a question. A problem is normally be accompanied by a detailed description.

Claim: An assertion about a given problem situation that can be either supported or refuted. For example, “Cleanroom engineering provides a viable solution in producing zero defect software.” One important characteristic of a **claim** is *falsifiability*, i.e., a claim can be falsified through evidence and logical reasoning.

Depending upon the level of evidentiality, claims can be divided into four types: *conjecture*, *hypothesis*, *fact*, and *axiom*:

- *Conjecture*: an omen-like claim with little, if any at all, supporting evidence. Commonly, it is a result of “scientific imagination,” and seen in highly theoretic domains, such as mathematics.
- *Hypothesis*: a claim that is derived from an existing theory or based upon available empirical/experiential evidence. Hypotheses formulation and testing constitutes a substantial portion of scientific activities. Hypotheses, when well supported by empirical evidence, are considered as scientific “facts.”
- *Fact*: a claim that is well supported by available evidence, and thus considered as true based on the current state of knowledge.
- *Axiom*: a claim that is generally accepted as true, and thus may be used as a basis for inference or argument.

The most common form of claims in the scientific discourse is hypothesis. A claim is always made in relation to a particular problem, regardless whether it is explicitly stated.

Evidence: Data gathered for the purpose of supporting or refuting a claim. For example, “The use of cleanroom techniques has yielded a 10-fold reduction of defects in the project Alpha.” Evidence can be either qualitative or quantitative. It is also always stated with respect to a given claim, and used to support or counter the same claim.

Theory: A systematic formulation about a particular phenomenon or problem. An example of **theory** is Halstead’s theory of software complexity, i.e., software science. A theory normally consists of a set of coherent, inter-related claims about the targeted problem domain. These claims may be derived deductively from other theories, or inductively

from well-established facts or hypotheses. An important characteristic of the theory is its *predictability*; a theory can be used to predict the outcome of a particular phenomena.

Concept: A primitive construct used in formulating theory, claim, or method. Examples of **concept** are: “meta-learning,” “knowledge representation,” “example-based learning.” The main feature of the concept is atomicity: concepts are the basic building block of human knowledge.

Method: Procedures or techniques used for generating evidence for a particular claim. Examples of **method** are: Delphi study, nominal grouping technique, Waterfall model of system development. A method itself can be the target of an inquiry.

Source: An identifiable written artifact, either artifact itself or the pointer to it, i.e., surrogate or reference. Examples of **source** are: “War and Peace,” “Slides from John’s presentation,” and so on. A source is a *medium* which contains other conceptual constructs, such as **problem**, **theory**, **concept**. However, **source** it in itself is not a conceptual construct.

Thing: An object, event, or process (natural or man-made) that is the source of a problem or target of an inquiry. Examples are “Unix,” “Macintosh,” “CLARE.”

Other: An open-ended node primitive that can be used to represent anything that falls outside of the above node primitives.

Summarative link primitives

Addresses: This link specifies a “carrier” relationship between an artifact (**source**) and a conceptual theme (**problem**). It consists of one tuple: **source** $\xrightarrow{\text{addresses}}$ **problem**. For example,

- Mill92: “Certifying the Correctness of Software” by Harlen Mill.
- Inadequacy of unit testing: Unit testing and debugging cannot uncover and remove all important errors in complex software system.
- Hence, **Mill92** $\xrightarrow{\text{addresses}}$ **Inadequacy of unit testing**.

Responds-to: This link describes a knowledge-level “stimulus/response” relationship between two RESRA nodes: the “stimulus” is the uncertain, perplexing state of knowledge about a particular phenomena, while the “response” is a position or assertion that aims at clarifying or resolving that situation. The tuple is expressed as **claim** $\xrightarrow{\text{responds-to}}$ **problem**, For example,

- Inadequacy of unit testing: Unit testing and debugging cannot uncover and remove all important errors in complex software system.
- Cleanroom reduces defects: Zero defect software is an achievable goal by using rigorous development and formal verification techniques from cleanroom engineering.
- Hence, **Inadequacy of unit testing** $\xrightarrow{\text{responds-to}}$ **Cleanroom reduces defects**.

Suggests: This link represents a “triggering” relationship, i.e., the presence of one RESRA instance leads to the recognition or awareness of the other. It consists of four tuples:

- $\boxed{\text{claim}} \xrightarrow{\text{suggests}} \boxed{\text{problem}}$
- $\boxed{\text{theory}} \xrightarrow{\text{suggests}} \boxed{\text{problem}}$
- $\boxed{\text{theory}} \xrightarrow{\text{suggests}} \boxed{\text{claim}}$
- $\boxed{\text{evidence}} \xrightarrow{\text{suggests}} \boxed{\text{problem}}$

See Section 3.3.3 for additional tuples. Below is an example of *suggests*:

- Cleanroom engineering minimizes defects: Zero defect software is an achievable goal by using rigorous development and formal verification techniques from cleanroom engineering.
- difficulty in achieving zero defect UI: A defect user interface is not only difficult to use but also difficult to verify using formal techniques. Hence, cleanroom engineering may not be the solution.

- $\boxed{\text{Cleanroom engineering minimizes defects}} \xrightarrow{\text{suggests}} \boxed{\text{difficulty in achieving zero defect UI}}$

Presupposes: This link depicts a logical dependency relationship between two claims,

i.e., $\boxed{\text{claim}_i} \xrightarrow{\text{presupposes}} \boxed{\text{claim}_j}$. For example,

- Programming as theory-building: Programming is “theory-building” (Peter Naur).
- Reconstruction of a programming theory: Re-establishing the theory of a program merely from the documentation is strictly impossible.
- Hence,

$\boxed{\text{Reconstruction of a programming theory}} \xrightarrow{\text{presupposes}} \boxed{\text{Programming as theory-building}}$

Is-alternative-to: This link describes a *competing* relationship between two RESRA nodes of the same type. It consists of four tuples:

- $\boxed{\text{Problem}_i} \xrightarrow{\text{is-alternative-to}} \boxed{\text{problem}_j}$
- $\boxed{\text{Claim}_i} \xrightarrow{\text{is-alternative-to}} \boxed{\text{claim}_j}$
- $\boxed{\text{Method}_i} \xrightarrow{\text{is-alternative-to}} \boxed{\text{method}_j}$
- $\boxed{\text{Theory}_i} \xrightarrow{\text{is-alternative-to}} \boxed{\text{theory}_j}$

Below is an example of the alternative claim:

- unrecoverable theory: Re-establishing the theory of a program merely from the documentation is strictly impossible.
- documentationist: Improved methods of documentation are able to communicate everything necessary for the maintenance and modification of a program.
- Hence, $\boxed{\text{unrecoverable theory}} \xrightarrow{\text{is-alternative-to}} \boxed{\text{documentationist}}$.

Defines: This link describes a formative relationship between a more complex RESRA node and a more singular one. It consists of three tuples:

- $\boxed{\text{claim}} \xrightarrow{\text{defines}} \boxed{\text{concept}}$
- $\boxed{\text{claim}} \xrightarrow{\text{defines}} \boxed{\text{method}}$
- $\boxed{\text{theory}} \xrightarrow{\text{defines}} \boxed{\text{concept}}$

Below is an example of *defines*:

- CLARE's Claim: CLARE provides an effective means of exposing differences and similarities among individual views on a given research paper.
- RESRA: RESRA is a language specifically designed for representing the thematic feature of learning artifacts and individual learners' views on them.
- Hence, $\boxed{\text{CLARE's Claim}} \xrightarrow{\text{defines}} \boxed{\text{RESRA}}$

Strengthens: This link describes a supporting relationship between a claim and a theory,

i.e., $\boxed{\text{claim}} \xrightarrow{\text{strengthens}} \boxed{\text{theory}}$.

Weakens: This link describes a countering relationship between a claim and a theory,

i.e., $\boxed{\text{claim}} \xrightarrow{\text{weakens}} \boxed{\text{theory}}$

Below is an example:

- Halstead's software science: A theory of software complexity.
- Milliman's claim: Milliman found that complexity measures of Halstead are better predictors of time to find the source of a fault than is the number of lines of code (LOC).
- Hence, $\boxed{\text{Milliman's claim}} \xrightarrow{\text{strengthens}} \boxed{\text{Halstead's software science}}$.

Supports: This link represents that the evidence is congruent with the directionality of

the prior claim, i.e., $\boxed{\text{evidence}} \xrightarrow{\text{supports}} \boxed{\text{claim}}$.

Counters: This link shows that the evidence points in a direction different from the prior claim: $\boxed{\text{evidence}} \xrightarrow{\text{counters}} \boxed{\text{claim}}$. For example,

- Unrecoverable theory: Re-establishing the theory of a program merely from the documentation is strictly impossible;
- documentationist: Improved methods of documentation are able to communicate everything necessary for the maintenance and modification of a program.
- T_EX experience: Hundreds of T_EX users around the world have demonstrated they understand the “theory” of the T_EX program through making special-purpose extensions to the existing code and by offering highly appropriate advice to users, even though they have only read its documentation.
- Hence,

- $\boxed{\text{T}_{\text{E}}\text{X experience}} \xrightarrow{\text{counters}} \boxed{\text{Unrecoverable theory}}$

- $\boxed{\text{T}_{\text{E}}\text{X experience}} \xrightarrow{\text{supports}} \boxed{\text{documentationist}}$

Generates: This link describes a causal relationship between the use of a given method and the outcome. Its tuple form is $\boxed{\text{method}} \xrightarrow{\text{generates}} \boxed{\text{evidence}}$. For example,

- Fagan’s Inspections: JPL adopted the 7-step Fagan’s Inspection method with checklist of tailored questions. Extensive training were provided to both managers (in the value of inspections) and developers (to get most out of inspections).
- JPL’s success: Within the period of 21 months, 300 inspections have been conducted; 10 projects have adopted the method as part of their procedures. The number of defects per inspection were 4 major defects and 12 minor ones. The average cost for finding, fixing, and verifying a defect is between \$9 and \$12 , compared to about \$10,000 to find and fix the same defect in the later life cycle.

- Hence, Fagan's Inspections $\xrightarrow{\text{generates}}$ JPL's success

3.3.2 Evaluative primitives

The RESRA's evaluative node and link primitives are intended for representing the learner's assessments and subjective views on the current artifact. Unlike their summarative counterparts, which aim at answering the question, "what do you think this paper is about?" the evaluative primitives are used to answer the question, "what do you think about this paper?"

RESRA provides three evaluative node primitives: critique, question, and suggestion. Figure 3.2 depicts graphically the relationships among these primitives.

Evaluative node primitives

Critique: Critical remarks or comments about a particular claim, evidence, method, source, et al., or relationships between them. For example, "the example applications of cleanroom engineering so far have been limited to well-defined domains."

Question: Aspects of a claim, theory, concept, etc., or relationships between them, about which the current learner is still in doubt or does not understand. For example, How does box-structured design differ from object-oriented approach?

Suggestion: Ideas, recommendations, or feedbacks on how to remedy or improve an existing problem formulation, claim, method, et al. For example, "I would like to see cleanroom engineering used in some non-conventional domains, such as groupware."

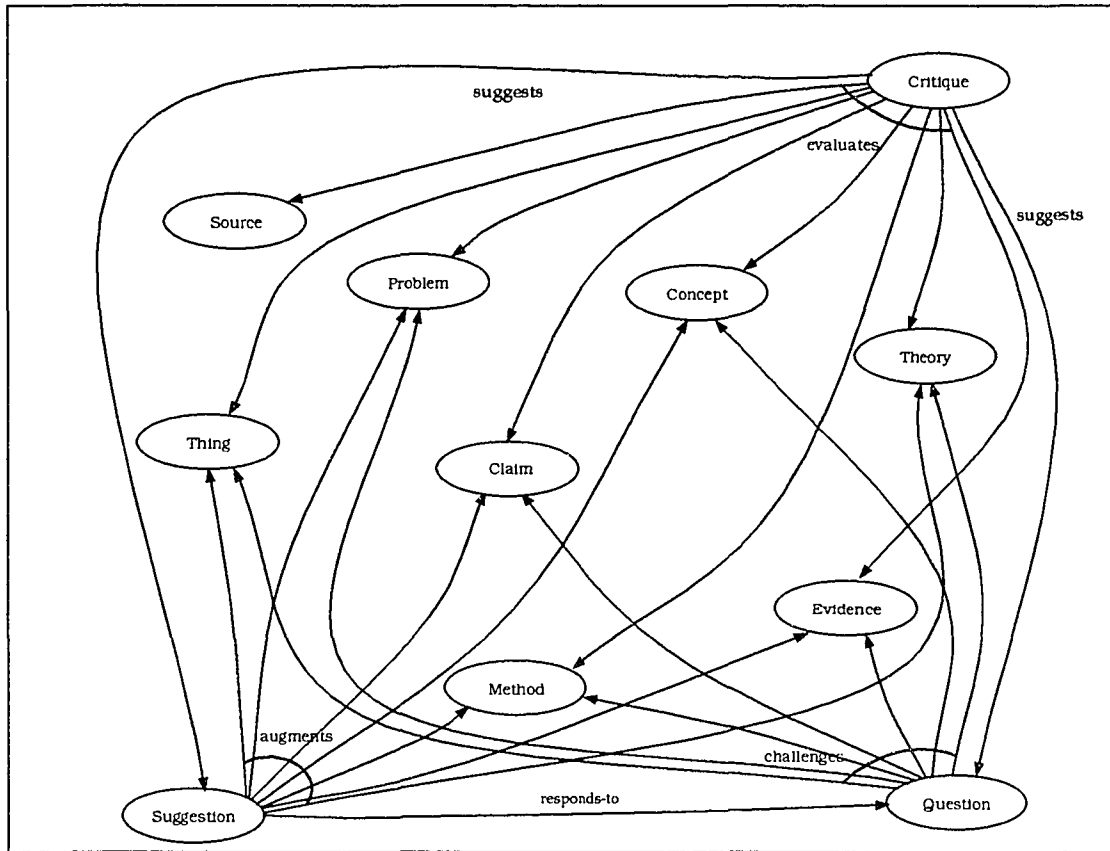


Figure 3.2: RESRA evaluative primitives at a glance

Evaluative link primitives

Evaluates: This link associates a critical comment with its target, i.e., `critique` $\xrightarrow{\text{evaluates}}$

`RESRA node`¹. For example,

- **Claim:** Zero defect software is an achievable goal by using rigorous development and formal verification techniques from cleanroom engineering.

¹ `RESRA node` stands for all currently defined RESRA node types, including critique itself.

- Critique: All three applications quoted in the paper are “well-structured” problems, i.e., problems whose functional and performance parameters can be precisely specified. Cleanroom engineering might run into problems with less well-structured domains, such as knowledge-based systems, collaborative software.

- Hence, critique $\xrightarrow{\text{evaluates}}$ claim

Challenges: This link directs a question to its target, i.e., question $\xrightarrow{\text{challenges}}$ RESRA node. For example,

- Question: What is the difference between box structured software design and object-oriented design?
- Concept: Box structured design is based on a Parnas usage hierarchy of modules. Such modules, also known as data abstractions or objects, are described by a set of operations that may define and access internally stored data.

- Hence, question $\xrightarrow{\text{challenges}}$ concept

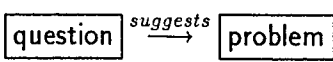
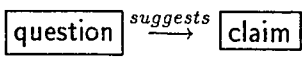
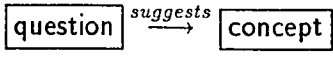
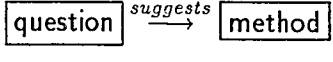
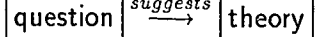
Augments: This link describes a remedial relationship between a proposal and an existing situation, i.e., suggestion $\xrightarrow{\text{augments}}$ RESRA node. For example,

- Suggestion: I would like to see cleanroom engineering used in some non-conventional domains, such as groupware
- Claim: Zero defect software is an achievable goal by using rigorous development and formal verification techniques from cleanroom engineering.

- Hence, Suggestion $\xrightarrow{\text{augments}}$ claim

3.3.3 Argumentative primitives

RESRA argumentative primitives are used for deliberating alternative positions and points of view on a particular problem or artifact. One main feature of CLARE is that it treats artifact-mediated argumentation among a group of researchers, e.g., debates taking place through a series of exchanges of artifacts, in the same way as the argumentation taking place within a classroom among a group of students. It uses RESRA for both purposes. The argumentative primitives are the aggregation of summarative and evaluative ones described in Sections 3.3.1 and 3.3.2, plus the following tuples:

- 
- 
- 
- 
- 

RESRA argumentative primitives are depicted in Figure 3.3. The solid lines represent newly-added links.

3.3.4 Integrative primitives

Integrative primitives are used for consolidating and inter-relating RESRA instances created by different learners. The focus of integration is on the relationship among existing nodes. Hence, it does not lead to the creation of new nodes, except for annotations, which

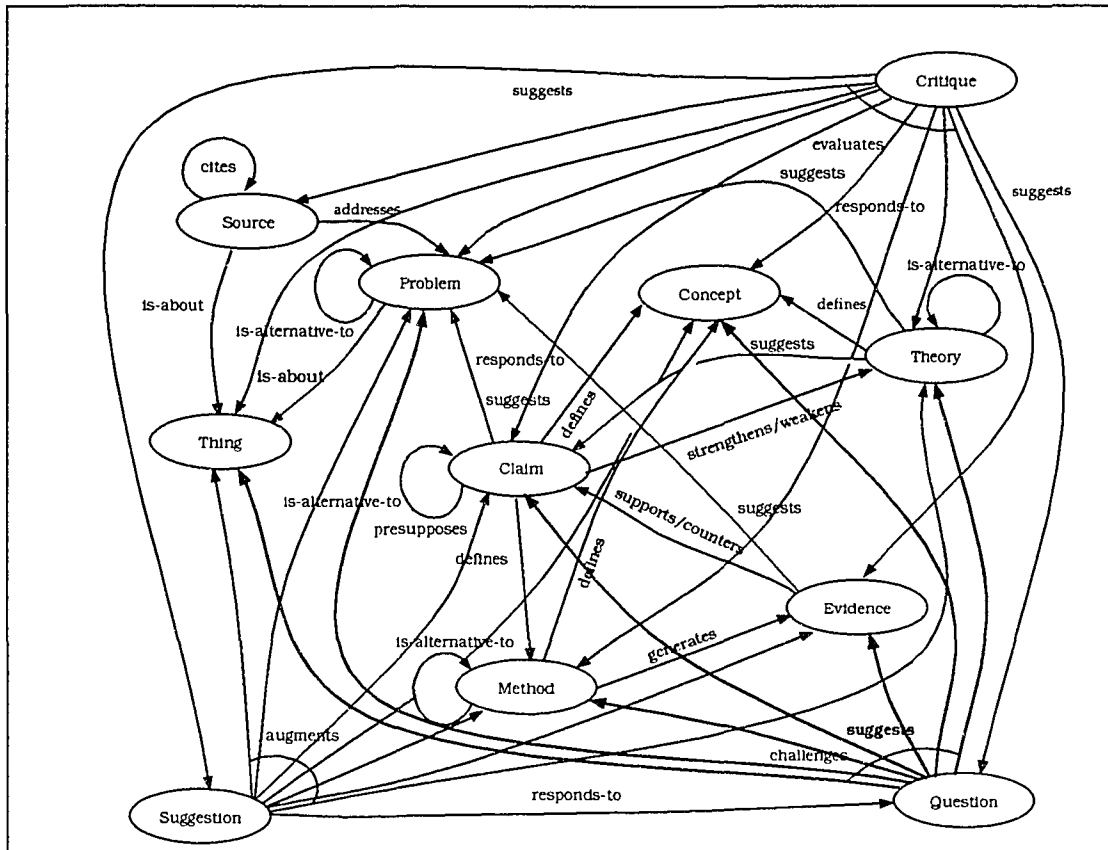


Figure 3.3: RESRA argumentative primitives at a glance

document the reasons behind endorsing a particular relationship. RESRA has four integrative link primitives, which are described below.

Is-similar-to: This link declares the two RESRA node instances share the same view and thus may be consolidated into one. Automatic merging of two nodes are not always possible or desirable, since it is rare that two nodes are totally congruent. The *is-similar-to* relationship is restricted to the RESRA nodes of the same type. In other words, one cannot declare that a `concept` *is-similar-to* a `claim`, or vice versa. However, one can declare one

`concept` as *is-similar-to* another `concept`. From one of the previous examples, one may declare:

`Box structured design` $\xrightarrow{\text{is-similar-to}}$ `OO design`

Share-same-perspective: A perspective represents a consistent way of viewing a problem or phenomena. Learners can share the same perspective, even though the detailed node and link instances they create are different. For example, important views on a given software project might include:

- *Customer's/user's perspective:* Ensuring their needs for a new system being met;
- *Management perspective:* Focusing on project management issues, including cost, deadline, employee morales, etc.;
- *Designer's perspective:* Ensuring that the requirement specification is realized through optimal algorithms and high reliability; and
- *Programmer's perspective:* Ensuring that the design is implemented consistently, efficiently, and readably (through proper documentation).

A RESRA node instance on an experience report of a software project is likely to fall into one of the above perspectives. By aggregating RESRA nodes by perspectives, it introduces necessary structures that help make the group outcome more understandable.

Contains: This link represents a part-whole relationship between two RESRA nodes. Typically, the two connected nodes should be the same type, though it is not always necessary. For example, CLARE is a thing, while RESRA is a concept. Nevertheless, one can

declare `CLARE` $\xrightarrow{\text{contains}}$ `RESRA`

Is-related-to: This link is open-ended, and can be used to express any type of relationship that falls outside the above categories. As a result, it should be used only when no more specific type is applicable.

3.4 Canonical RESRA Forms (CRFs)

3.4.1 Overview

The Canonical RESRA Form (CRF) is a *template* of RESRA node and link primitives used to represent the typical thematic structure of scientific text. It represents commonly accepted formats by which the practitioners in a given domain formally communicate and share knowledge. CRFs may be viewed in terms of Kuhnian notion of the *paradigm*) of the scientific practice. As the newcomer to a field, the student may find it helpful to learn about the major CRFs used in that domain, and to be able to recognize them in their various disguised forms, apply them in evaluating existing artifacts and in constructing their own artifacts, and use them to help organize domain knowledge and collaborate with their peers.

Because of the unique nature of the problems in each subject domain and the concomitant methodological differences, CRFs often vary from one field to another. For example, software engineering, which is a relatively young field, exhibits a variety of relatively ill-defined templates. In contrast, psychology, perhaps one of the most well-developed social science disciplines, contains a much smaller number of less varied, more formalized templates. This document provides a set of templates derived from the domains of software engineering, in particular, software quality assurance (SQA). It is important to note that, like RESRA, CRFs presented below are exemplary rather than exhaustive. The learners are encouraged to use these examples as a basis for creating their own domain-specific CRFs.

3.4.2 Five example CRFs

Each of the following CRFs consists of a short description, a graphical representation, and an example to illustrate how the CRF might be applied to an artifact.

Concept papers

Description: A *concept paper* presents a new conceptualization of a problem, or a new method, technique, or approach to addressing an existing problem. In a concept paper, the author might back up his claims by some type of evidence, such as contrived examples, experiential or empirical data, and so on. However, the main contribution of such a paper does not lie in the strength of its supporting data but rather the demonstrated novelty and potentials of the proposed technique, concept, etc.

Figure 3.4 shows the CRF for concept papers. At the core of a concept paper lies the *claim* about how the new *method* and *concept* described in the paper help solve the *problem* conceived by the author. Even though some *evidence* might be provided to support the claims, the major portion of the paper is typically devoted to the definition of new concepts, and the elaboration of the method and the relationship between the two. The claim might be broken down into sub-claims that are treated separately.

An example: A good example of a concept paper is [JT93], whose RESRA node representation is listed below. Figure 3.5 provides a complete RESRA representation. Note that the nbuff review example is used to illustrate various aspects of the method, although it might also be considered as the evidence for supporting the authors' claims. The main contribution of the paper, however, does not lie in the supporting evidence, which is quite weak, but rather in the way in which the problem is presented and the way in which the method is related to the problem.

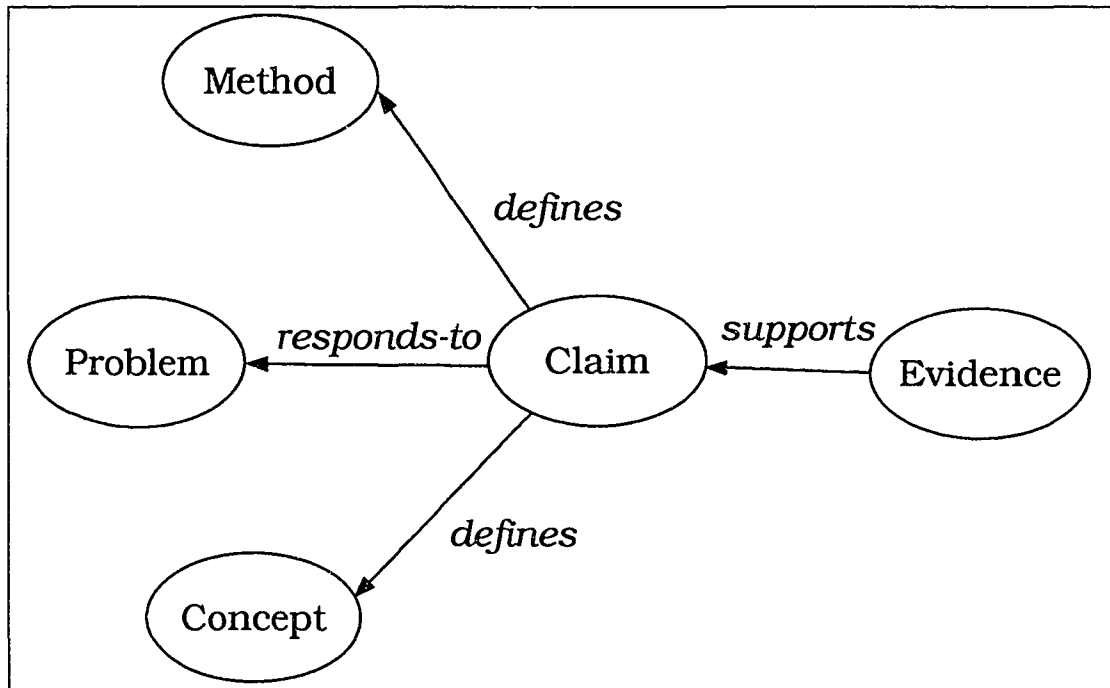


Figure 3.4: A CRF for concept papers

Problem: There exist three main road-blocks to fully effective formal technical review: labor-intensive nature of review, incompatibility with incremental development methods, and lack of support by existing tools for adapting review methods to specific organizational contexts.

Claim: CSRS provides an effective means to overcome the above three barriers and to realize full potentials of FTR.

Method: The CSRS data and process models.

Evidence: Review of NBUFF using CSRS.

Experience Papers

Description: An *experience paper* is a factual account of the experience of an individual, group, or organization with a new method, technology, instrument, theory, etc. An

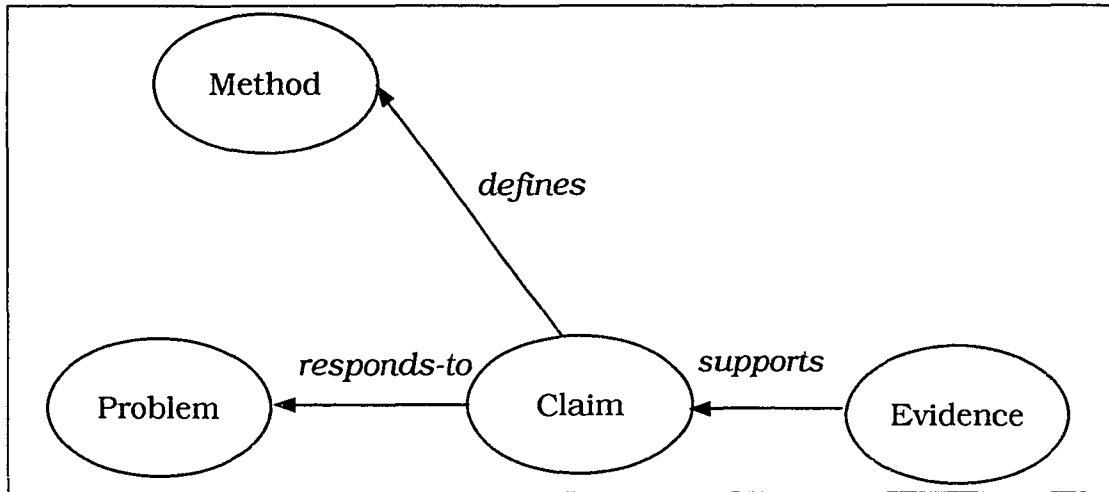


Figure 3.5: A RESRA representation of [JT92]

experience paper does not have the same level of rigor as an empirical study or novelty as a concept paper. The important contribution of an experience paper lies in the fact that it is from the real world. Since any model, theory, claim, or technique requires some level of generalization, it needs change when applied back to a specific real world situation. The experience paper is often used to report such experience.

Figure 3.6 shows a CRF for experience papers. Such a paper typically begins with a problem confronted by an individual, group, or organization. Based upon previous work or the experience from other people, a claim is made that a given method/technique should be used to solve the problem. The method is then applied and the experience is described. Like empirical studies, the outcome from an experience paper may be either positive or negative; both are equally valuable. In the case of failure, what is important is the discussion of the possible cause of failure and lessons learned.

An example: A good example of the experience paper is [Bus90], which discusses the experience of introducing formal software review to JPL. The RESRA node instances for

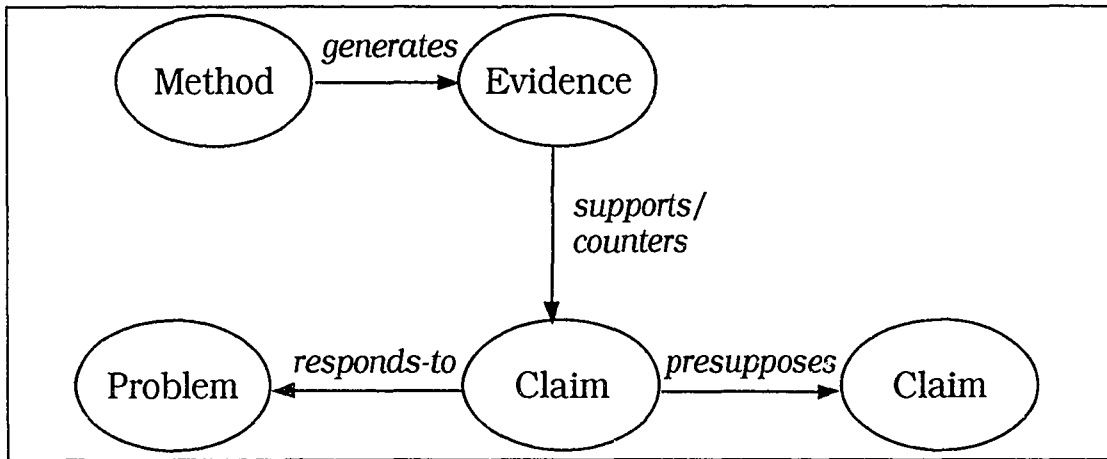


Figure 3.6: A CRF for experience papers

this paper are listed below. The relationships between those nodes are depicted in Figure 3.7.

Problem: Software has become a rising part of the JPL work hours (from 50% of the mid-1980's to estimated 80% by the year 0% 2000) and, as a result, so is the importance of being able to produce error-free systems in a cost-effective manner.

Claim₁: Fagan reports that the use of the Fagan's Inspections can correctly find up to 95% of all defects before entering the test phase.

Claim₂: Fagan's method is the most cost-effective defect-detection technique appropriate to the JPL.

Method: JPL adopted the 7-step Fagan's Inspection method with checklist of tailored questions. Extensive training were provided to both managers (in the value of inspections) and developers (to get most out of inspections).

Evidence: Within the period of 21 months, 300 inspections have been conducted; 10 projects have adopted the method as part of their procedures. The number of defects per inspection were 4 major defects and 12 minor ones. The average cost for finding, fixing, and verifying a defect is between \$9 and \$12 , compared to about \$10, 000 to find and fix the same defect in the later life cycle.

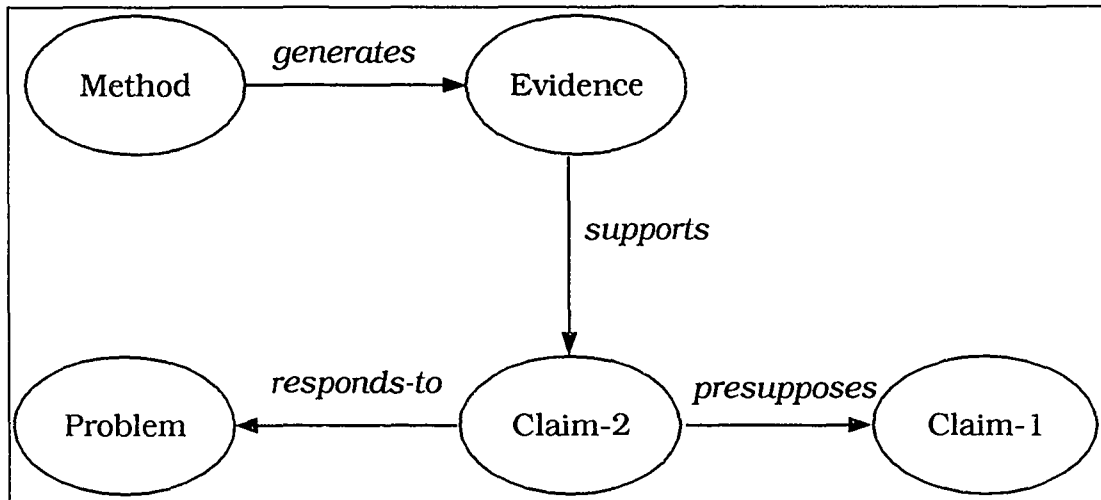


Figure 3.7: A RESRA representation of [Bush, 90]

Empirical papers

Description: An *empirical paper* reports the result of a hypothesis-driven, systematic investigation of particular problem domain. It involves experimentation, i.e., hypothesis formulation, experimental design, execution, data analysis, and drawing conclusions. Unlike the concept paper, which introduces new methods or new formulation of a problem, empirical studies often attempt to demonstrate whether an existing method or understanding of a problem is indeed well-grounded. Hence, the empirical study starts at where the concept paper leaves off. Empirical studies constitute a major percentage of artifacts in established disciplines. This is not surprising given the fact that a mature field typically have methods and problems defined.

Figure 3.8 shows CRF for empirical papers. The beginning and end point of an empirical study is the claim, or hypothesis, which is either derived from a theory in a field or based on the claims made in previous work. The key to an empirical study is the lower triangle, i.e., claim, method, and evidence. The methodology plays a vital role in this process because it directly affects the validity and generalizability of the data and the outcomes.

The resulting data may or may not support the initial hypothesis, however. This is quite different from the concept paper in which only positive data is reported.

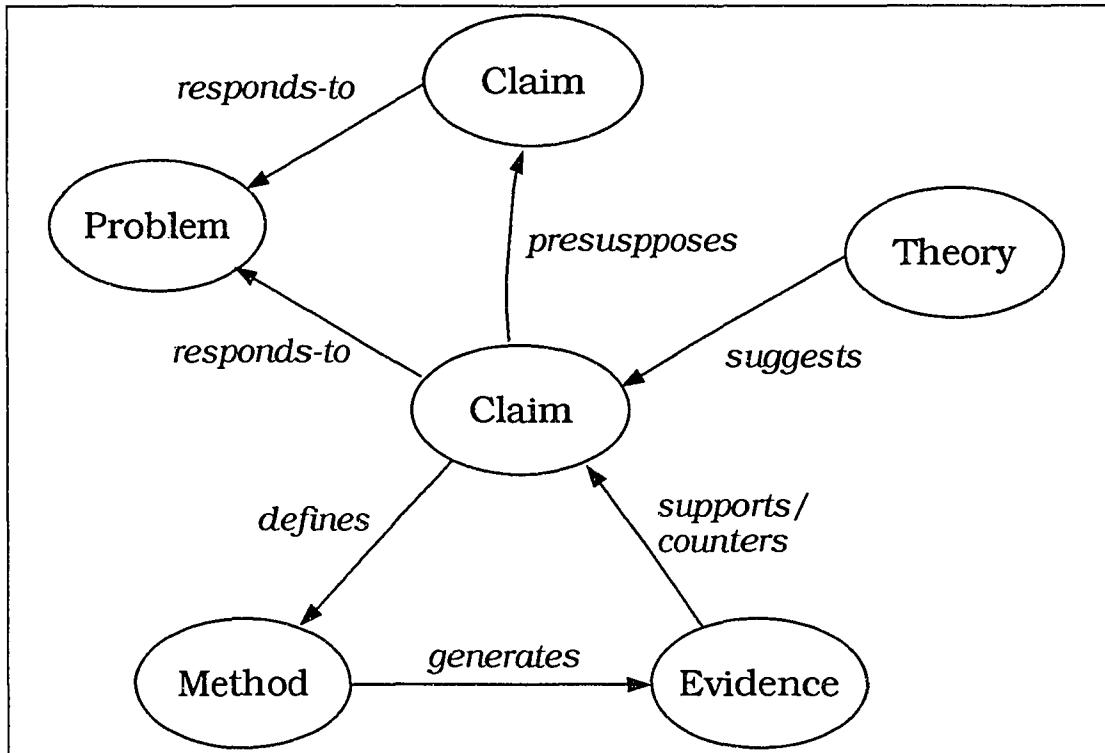


Figure 3.8: A CRF for empirical papers

An example: [CSM79] is a typical empirical paper, which reports the result of the third experiment attempting to relate the complexity metrics developed by Halstead and McCabe to the difficulty programmers experience in understanding and modifying programs. The RESRA node instances are shown below, and the relationships between them are depicted in Figure 3.9. Note that the RESRA representation of this paper matches only with the lower triangle in Figure 3.8.

Problem: Software complexity metrics are abound. However, studies thus far are still inconclusive about which metric is the best predictor of programmer performance.

Claim: Software complexity metrics that count operators, operands, and elementary control flows are better predictors of the difficulty programmers experience in working with software than a simple count of LOC.

Method: Fifty-four professional programmers from six different locations participated in the experiment that involved three different programs, each with three different versions of control flow, and they were presented in three different length varying from 25 to 225 lines of code. To control for individual difference in performance, a within-subject 3^4 factorial design was employed.

Evidence: The result indicated that, at the subroutine level, all three complexity metrics (i.e., Halstead's E, McCabe's $v(G)$, and LOC) predicted the performance equally well, accounting for 40-45% of the variance in performance scores. At the program level, however, Halstead's E accounted for over twice as much variance in performance as the LOC (56% vs. 27% respectively) while the variance accounted for by McCabe's $v(G)$ fell between these values (42%).

Essays

Description: The *essay* is a loosely-defined artifact, sometimes called the "opinion paper," which contains observations and elucidation of a particular problem or issue. Often, essays are written by experts in the field; their opinions and points of view represent insights distilled from their years of experience. Though methodologically they are not as *scientific* and rigorous, essays are an important part of the intellectual landscape in many disciplines, especially in applied ones such as software engineering, in which expertise plays a just as important, if not more important, role as the absolute truth.

Essays vary widely in styles. At the thematic level, however, their structures are quite singular, generalizable into some standard forms. Figure 3.10 shows one RESRA template of the essay paper. The center of such an artifact is often the expert's opinions or claims. The problem at which the claim is targeted may or may not be explicitly described. An-

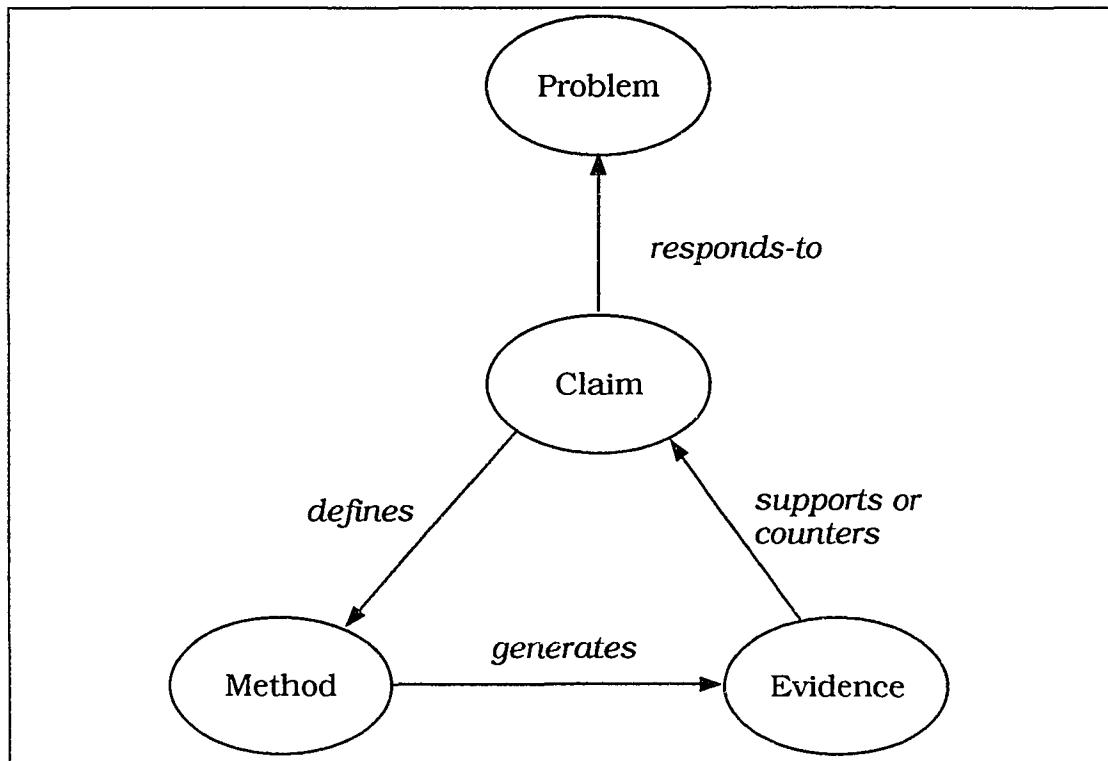


Figure 3.9: An example RESRA representation of [CSM79]

tithetical claims are typically identified and countered by the same evidence that supports the author's particular claim. Evidence in an essay is typically anecdotal, drawing heavily from the author's experience and observations, though it is not uncommon to see secondary evidence from other authors in the field. It is also uncommon that essays introduce new methods or concepts.

An example : [Knu92] is a good example of essays, whose RESRA representation is listed below. Figure 3.11 shows the relationships among the various nodes. Note that this paper as whole does not neatly fits to the template depicted in Figure 3.10, since it contains two groups of claims: primary (i.e., claim-1 and evidence-1 and secondary (i.e., the rest)).

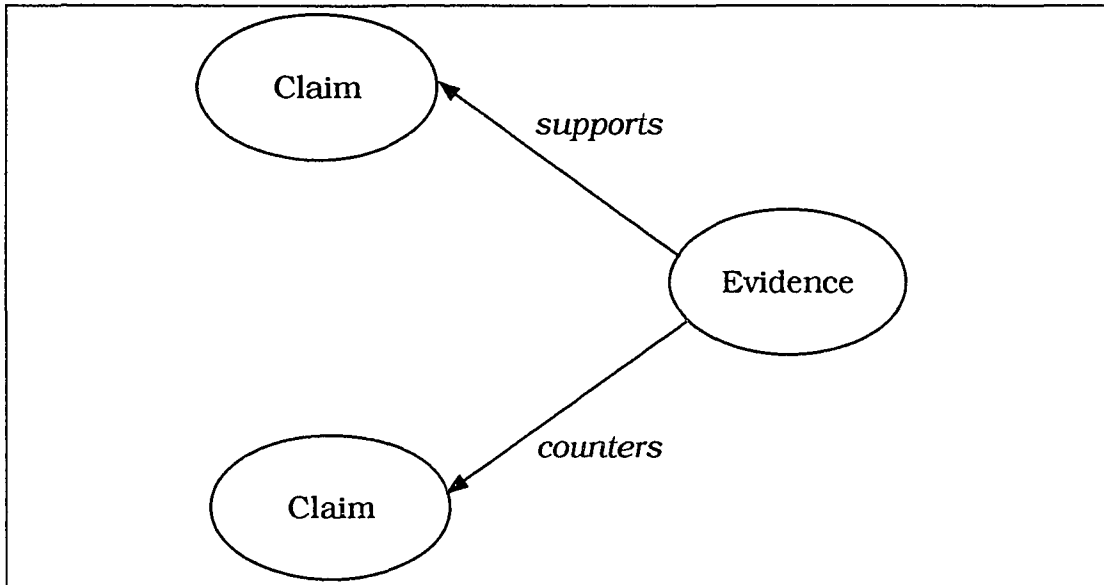


Figure 3.10: A CRF for research essays

However, the latter matches with the template. Such deviations are not uncommon in essays, for their structural forms are much less restrictive compared to other artifact types.

Claim₁: Learning through trial-and-errors can be enhanced by maintaining a record of the mistakes one has made.

Evidence₁: The author's finds his error log of the T_EX system, which contains 867 entries, so instructive that he publishes it so that other people can benefit from his experience.

Claim₂: Programming is “theory-building” (from Peter Naur).

Claim₃: Re-establishing the theory of a program merely from the documentation is strictly impossible;

Claim₄: Improved methods of documentation are able to communicate everything necessary for the maintenance and modification of a program.

Evidence₂: Hundreds of T_EX users around the world have demonstrated they understand the “theory” of the T_EX program through making special-purpose extensions to

the existing code and by offering highly appropriate advice to users, even though they have only read its documentation.

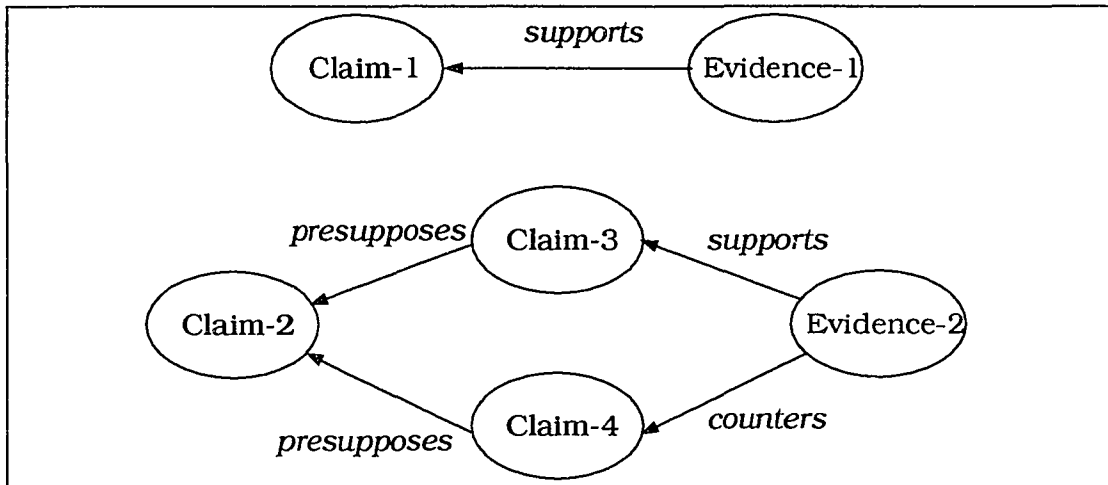


Figure 3.11: A RESRA representation of [Knuth92]

Survey papers

Description: A *survey* is an overview of existing work in a particular topical area or subject domain. It highlights what has been accomplished thus far, what major problems have been encountered, and where future efforts should be devoted to. A survey represents a *synthesis* of previous work. The main contribution of a survey paper is that it summarizes the current state of a given topic, and provides a single-point entry for the beginners of the field.

Figure 3.12 shows a CRF for survey papers. The focal point of a survey is the problem under concern. It brings together all related work, some of which may involve different claims, while others share the same claim but provide different evidence. At the end, the author normally attempts to summarize the above by offering his own concluding claims, in particular, about the direction in which the domain is heading.

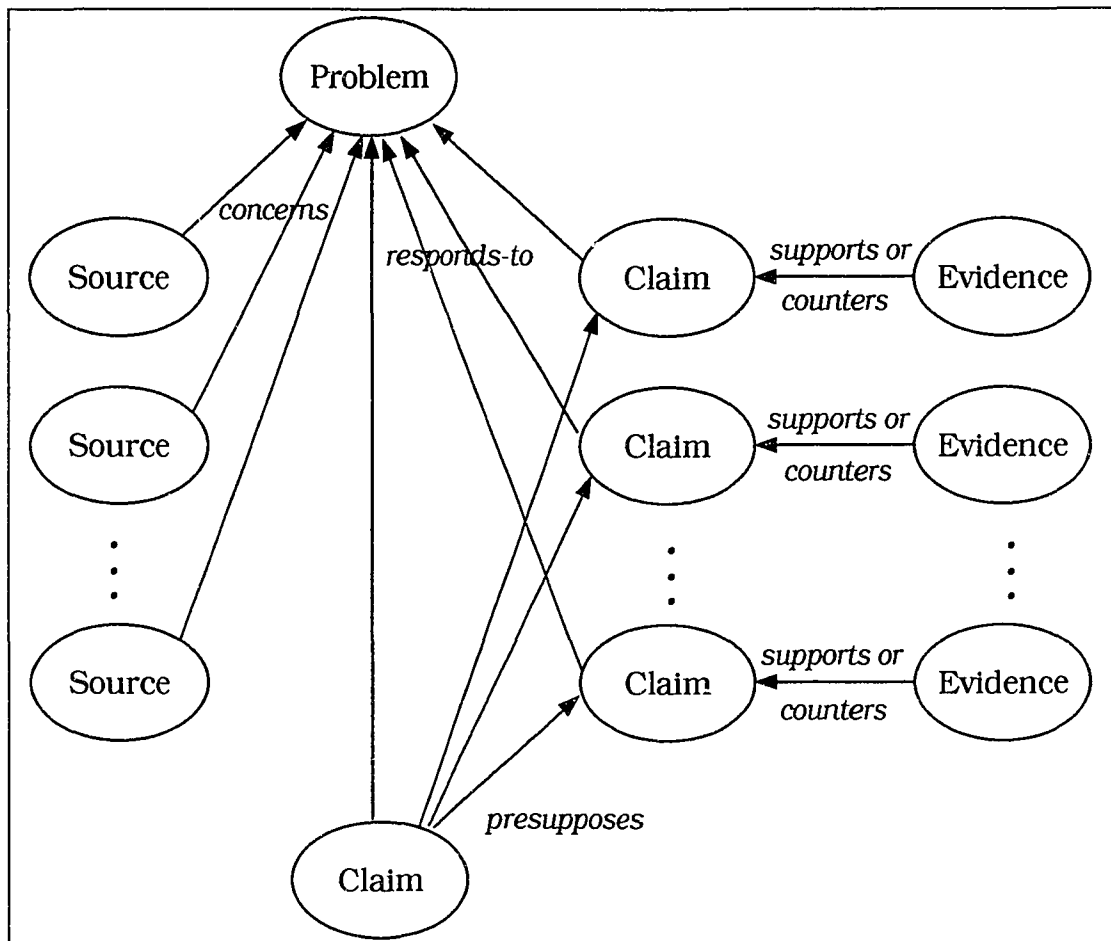


Figure 3.12: A CRF for survey papers

An example: An example survey paper is [Tic92]. The RESRA node instances for the paper are listed below, and the relationships between those nodes are depicted in Figure 3.13. Note that this paper is not very representative in that it covers an incremental but nevertheless singular view of the selected topic, i.e., the evolving concept of the “programming-in-the-large,” instead of multiple, competing views of the same topic, as one expects from a typical survey.

Problem: While programming-in-the-small is a well established discipline, programming-in-the-large, albeit no less important, is a much less developed subject area.

Source₁: Parmas, David. "Designing software for ease of extension and contraction." *CACM*, 15(2):1053-8, Dec'72.

Source₂: DeRemer, Frank and Hans, Kron. "Programming-in-the-large versus programming-in-the-small." *TSE*, SE-2(2):80-6, Jun'76.

Source₃: Tichy, Walter. "Tools for software configuration management." *In Proc. of Int. Conference on Software Version and Configuration Control*, Jan'88.

Claim₁: Parmas states that details that are likely to change during the evolution of a system should be made into separate modules, while the interface of those modules should be made insensitive to changes in these details (i.e., "information hiding.")

Claim₂: DeRemer and Kron claims that structuring a large collection of modules to form a "system" is an essentially distinct intellectual activity from that of constructing the individual modules (i.e., programming-in-the-small) and, correspondingly, distinct languages should be used for the two activities.

Claim₃: Software configuration management (SCM) improves programming-in-the-large by providing automated tool for such tasks as version control, configuration selection, and system building.

Evidence: Richkind's SCCS and Tichy's RCS for automated revision control; Feldman's make for maintaining, updating, and generating related programs.

Claim₄: The three promising measures for further improve quality and productivity in programming-in-the-large are: better education of software engineers; better tools; and avoiding programming or code reuse.

3.5 Extending RESRA

RESRA is designed to be an open-ended language. Regardless of the size of the initial primitive and template set, RESRA is incomplete with respect to the universe of artifacts

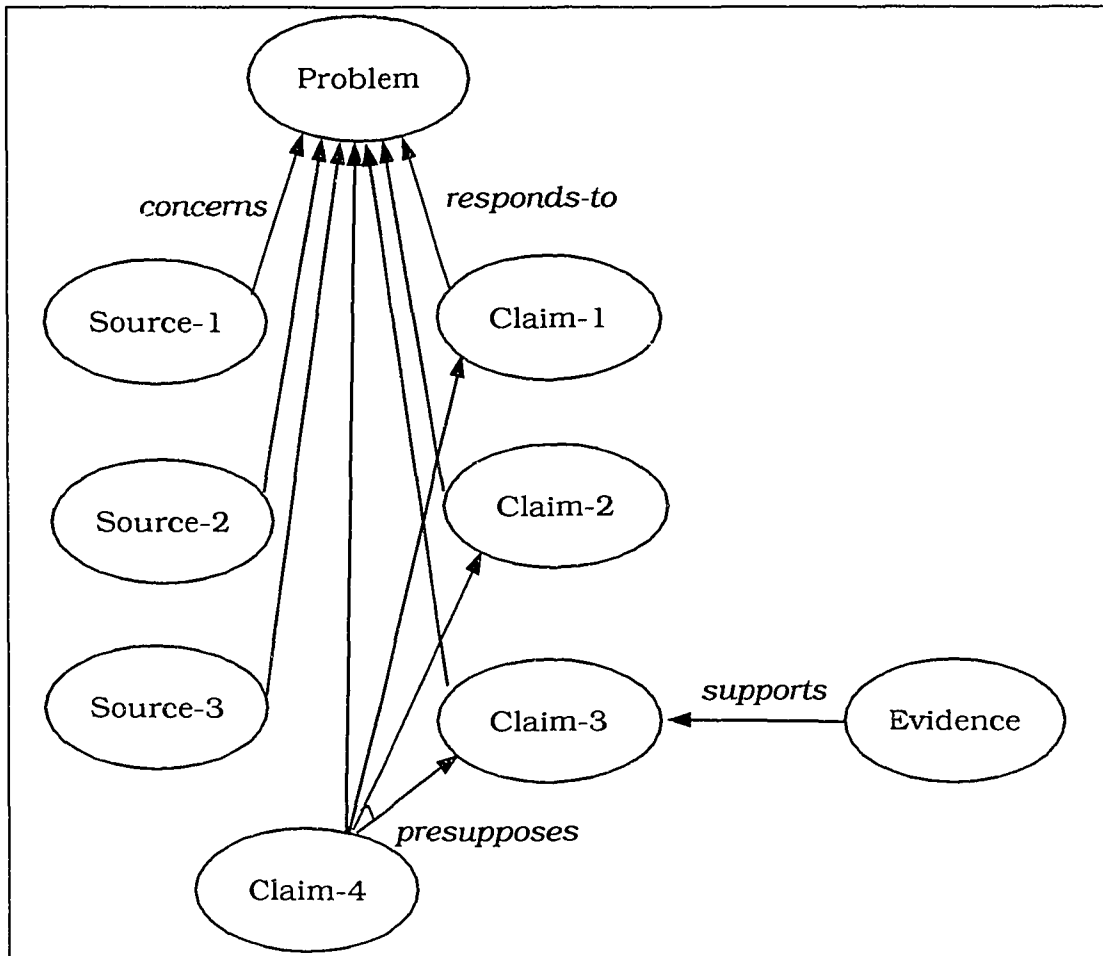


Figure 3.13: A RESRA representation of [Tichy, 92]

it is intended to represent. The goal is not, however, to provide a comprehensive, encyclopedic set of meta constructs for human knowledge, if such attempt is plausible. Instead, constructing such a *thorough* set of RESRA constructs is an ongoing activity that can be most profitably done by the learners themselves, rather than by a third party. The primary objective of this research is to provide a representation whose predefined constructs are:

- Coherent, self-contained, and powerful enough that can be used to adequately represent both the thematic feature of the learning artifact and different learners' points of view; and
- A basis from which any group of learners may collaboratively derive or adapt their own domain-specific or even group-specific representations. From this perspective, RESRA is always *illustrative*, which provides the learner with a sense of what to do next.

Learning to extend RESRA by creating new primitive types and canonical forms, or adapting existing ones is a significant, albeit not always easy, meta-level activity which CLARE aims at promoting. In fact, exploring the structural evolution of a representation is one of our original primary research interests. The underlying platform, i.e., EGRET, is designed to support such type-level changes [Joh91].

Chapter 4

CLARE: the System

CLARE is a computerized collaborative learning environment that implements the conceptual framework described in Chapter 2. This chapter provides an overview of the design and implementation of CLARE. Section 4.1 describes the requirements for the system. Section 4.2 discusses two major design considerations: a combined object-oriented and layered approach, and the initial choice of services over interface. Section 4.3 describes the five architectural components: *kernel*, *exploration*, *consolidation*, *initialization*, and *utilities*. Section 4.4 depicts the user interface. Section 4.5 shows a road map that outlines major process steps that the user goes through in a typical session. The chapter concludes with a brief review of the development history of CLARE, and a report of its current status.

This chapter is supplemented by a detailed design specification [Wan93a], and a user's guide to CLARE [Wan93b]. Readers whose primary interest is in using the system should read the latter. Readers with the system designer and implementor perspectives might find it worthwhile to browse the former. Readers with a research focus on CLARE should start with this chapter, and consult the other two documents as necessary.

4.1 Basic requirements

CLARE is a research tool. As such, it has three primary objectives: 1) to demonstrate the implementational viability of the proposed approach, that is, whether or not the approach can be implemented within the constraint of available resources; 2) to provide an instrumentation mechanism to capture the data necessary for gaining new insights on the nature

of computer-augmented collaborative learning; and 3) to show that CLARE is a usable system to the potential user population. The first objective is bounded by the conceptual framework of the current research; it is called the *conceptual* requirement. The second aims at yielding new empirical understanding of the underlying process; it is the *data* requirement of the system. And, finally, it is the *usability requirement*, which ensures the resulting system is usable by the intended user. The following sections address each of the three requirements.

4.1.1 Conceptual requirements

Conceptually, there are three key requirements for CLARE: (1) multi-user, distributed environment; (2) support for the RESRA representation language; and (3) support for the proposed collaborative learning model.

As a collaborative system. As a collaborative system, CLARE must support the following five essential functions:

- To allow multiple simultaneously connected users from physically dispersed locations;
- To ensure data consistency and integrity via concurrency control mechanisms;
- To maintain *What-You-See-Is-What-I-See* (WYSIWIS) among distributed clients, i.e., synchronization of the client's views of the shared data;
- To provide capabilities to identify individual users and track their status; and
- To manage access and actions on the shared data, for example, not allowing deletion of a node created by another user.

CLARE is primarily an asynchronous system. In other words, the *virtual co-presence* of multiple users in the system is not required for engaging online conversations. However, the effect from any user action ought to be communicated to all connected users in a consistent and transparent manner.

As a representation-based learning system. One key feature that differentiates CLARE from other collaborative learning systems is the embedded representation language, i.e., RESRA. In essence, CLARE is a RESRA-centered environment. Operations for manipulating RESRA primitives, templates, and instances constitute a core of that system. Specifically, CLARE must support the following capabilities:

- To define, update, annotate, instantiate, and view RESRA node and link primitives;
- To define, update, annotate, view, and associate RESRA templates with specific artifacts;
- To create, update, select, and navigate RESRA node instances;
- To support context-sensitive creation and deletion of link instances;
- To define, update, select, consult, and browse RESRA template instances;
- To define, update, and browse example RESRA node, link, tuple, and template instances;
- To support artifact-based comparison of node, tuple, and template instances among different users; and
- To generate hardcopy summary report of session activities.

The ability to extend RESRA by adding new node primitives, link primitives, and templates, and by amending existing ones is deemed essential to the long-term viability of RESRA. However, during an initial phase, such a feature does not have to be directly

available to the user via interactive invocation. This restriction gives the user opportunity to gain a full grasp of the existing primitives and templates before attempting to invent new ones.

As a collaborative learning system. The term *collaborative learning* has a well-defined meaning in CLARE: the use of RESRA to represent, evaluate, deliberate, and integrate the thematic features of scientific text according to the SECAI learning model (see Section 2.3.2 for details on SECAI). Below are a set of specific requirements derived from this model:

- RESRA-based operations should be structured into five groups, which correspond to the five components of SECAI: *summarization*, *evaluation*, *comparison*, *argumentation*, and *integration*;
- CLARE's main functions should be partitioned into at least two main modules, corresponding to the two phases of the SECAI model, i.e., *exploration* and *consolidation*;
- Since exploration is a *private* activity, CLARE needs to provide access control to ensure that a user at this phase is not allowed to view or act on RESRA instances created by other users; and
- The *anchor point* for exploration is the source nodes that constitute the selected artifact; the anchor point for consolidation is the RESRA representations of the contents of the selected artifact and the points of view on the content from individual users.

The SECAI learning model is adaptable to support a variety of learning tasks. For example, with the omission of *summarization* and *integration*, the system can be used to critique peer's term papers or project proposals. Similarly, by importing an interesting thread from any newsgroup into CLARE, one may only need the *argumentation* functions to continue the discussion in CLARE. Such variations, however, go beyond the initial scope of this prototype.

4.1.2 Data requirements

As a research tool, CLARE needs to be equipped with built-in instrumentation mechanism which gathers fine-grained process data. Specifically, such a mechanism must meet the following four requirements:

- Instrumented data points should be available for all important CLARE operations;
- Each data point should include the name of the operation, the order and exact time of invocation, and the name of the user who invokes it;
- Data capturing should be done in a non-obtrusive manner, and should not incur substantial runtime overhead; and
- Certain measures should be taken to maximize the validity of the process data gathered, e.g., eliminating spurious elapsed time.

Ultimately, CLARE should have an analysis back-end which automatically extracts instrumented process and outcome data and exports them into a format importable to plot programs (e.g., gnuplot), and statistical packages (e.g., SAS, S Plus). Currently, it is adequate that analyses be done on an ad hoc basis.

4.1.3 Usability requirements

Despite its research orientation, CLARE was designed to be used by users in real classroom settings. As a result, it needs to meet some minimal usability requirements, including:

- Abiding by established interface design standards, for example, [Com87];
- Response time for interactive commands should be acceptable; and
- Examples should be provided whenever possible to show the user the proper way of using the system constructs and functions.

In the initial prototype, no attempt is made to offer graphical means of browsing and manipulating RESRA instances. Hence, the usability or user-friendliness of CLARE is ultimately constrained by the textual nature of its interface.

4.2 Main design considerations

CLARE is not a stand-alone system. Rather, it is a tool built upon a generic collaborative application platform called EGRET¹. The two principal design features of CLARE, namely, a layered, object-oriented design, and the emphasis of services over interface, are both inherited from the underlying platform.

4.2.1 Layered + object-oriented design

EGRET — the platform on which CLARE is built — was designed with the special consideration given to the extensibility and verifiability. To reach the two goals, it blends a layered design with the object-oriented method (see [JWT⁺93] for more details). CLARE has adopted the same design principle for two main reasons:

- CLARE is a domain-specific instantiation of EGRET. It relies heavily on EGRET for infrastructure support. As a result, the more consistent CLARE is with EGRET at the design level, the more it can leverage on the services EGRET offers, for example, the automatic generator of design documentation.
- Like in EGRET, extensibility is one of the primary concerns for CLARE. Because of the ill-structured nature of the problem domain, the initial implementation was pur-

¹EGRET, which stands for Exploratory GRoup EnvironmenT, is hypertext-based distributed environment for supporting collaborative applications. See [Joh94] for details on EGRET. Besides CLARE, there are also other applications built on top of EGRET, including CSRS — a collaborative software review system ([Tja93]), and URN — a collaborative knowledge structuring tool for USENET ([Bre93]).

posedly kept small, containing only functions that are well-understood and deemed as essential. As the usage experience of the system increases, and as the underlying process becomes better understood, new features, such as support for domain-specific RESRA, will be likely to find their way into the system. The combination of the layered and the object-oriented paradigm provides an effective means of addressing changing requirements.

See [Wan93a] for details of the object-oriented design specification of the system.

4.2.2 “Services” versus “interface”

Although having a fancy graphical user interface would probably enhance the usability of CLARE, there are two other competing design factors to consider as well. The first concerns the accessibility of the chosen platform. The state-of-the-art interface is typically available only from commercial vendors, some of which run only on specialized hardware and software platforms, for example, SGI or NextStep, which are not widely accessible. In contrast, publicly available environments, such as the ones from the Free Software Foundation, are often less up-to-date in terms of interface. At the very early stage, it was decided that both EGRET and CLARE should be implemented on a widely accessible platform. Hence, we have chosen what we consider as the best of publicly available environments, i.e., Lucid Emacs, as the initial delivery mechanism.

Second, CLARE is still an early stage of its evolution. It will be some time before it reaches the point of maturation. Until then, the primary purpose of this project remains as to explore the exact requirements for system through successive instrumented usage and revisions. The principal requirement at this phase is flexibility — the ability to accommodate new functionalities and to adapt the existing ones. The interface, on the other hand, occupies a secondary importance.

4.3 Architecture

Architecturally, the CLARE environment consists of two main components: the *platform* and the *system proper*. The former includes the EGRET client and the hypertext database server. Together, the two provide basic infrastructure-level services such as storage, retrieval, updating, and caching of typed nodes, links, and fields, data locking and synchronization, and so forth. The CLARE system proper is composed of five modules: *kernel*, *preparation*, *exploration*, *consolidation*, and *utilities*. The relationships between these five modules are depicted in Figure 4.1.

4.3.1 Kernel

The *kernel* performs two essential functions: it is an interface to EGRET, and it provides support for RESRA. As an interface, it encapsulates all primitive services provided by EGRET, including node and link creation, update, retrieval, packing/unpacking, display, attribute data caching, and etc. Hence, it forms the foundation of the entire system. All other CLARE modules need to go through the kernel to access necessary EGRET functions. This design provides a single point of entry to EGRET, which can minimize potential ripple effects due to changes in the underlying platform. In addition, it also enables flexibilities such as customized default parameters, additional error checking, etc.

The other principal service provided by the kernel is the support for RESRA representational language (see Chapter 3). Specifically, it include three types of operations: *primitives*, *CRFs*, and *examples*. Typical operations on RESRA primitives are initialization, instantiation, validity checking, type-based instance listing, and status tracking. Important CRF functions are CRF creation, updating, annotation, listing, consulting, and importing/exporting. The example mechanism allows the user to tag and untag the current node as an example, browse predefined examples in the current database, and import/export ex-

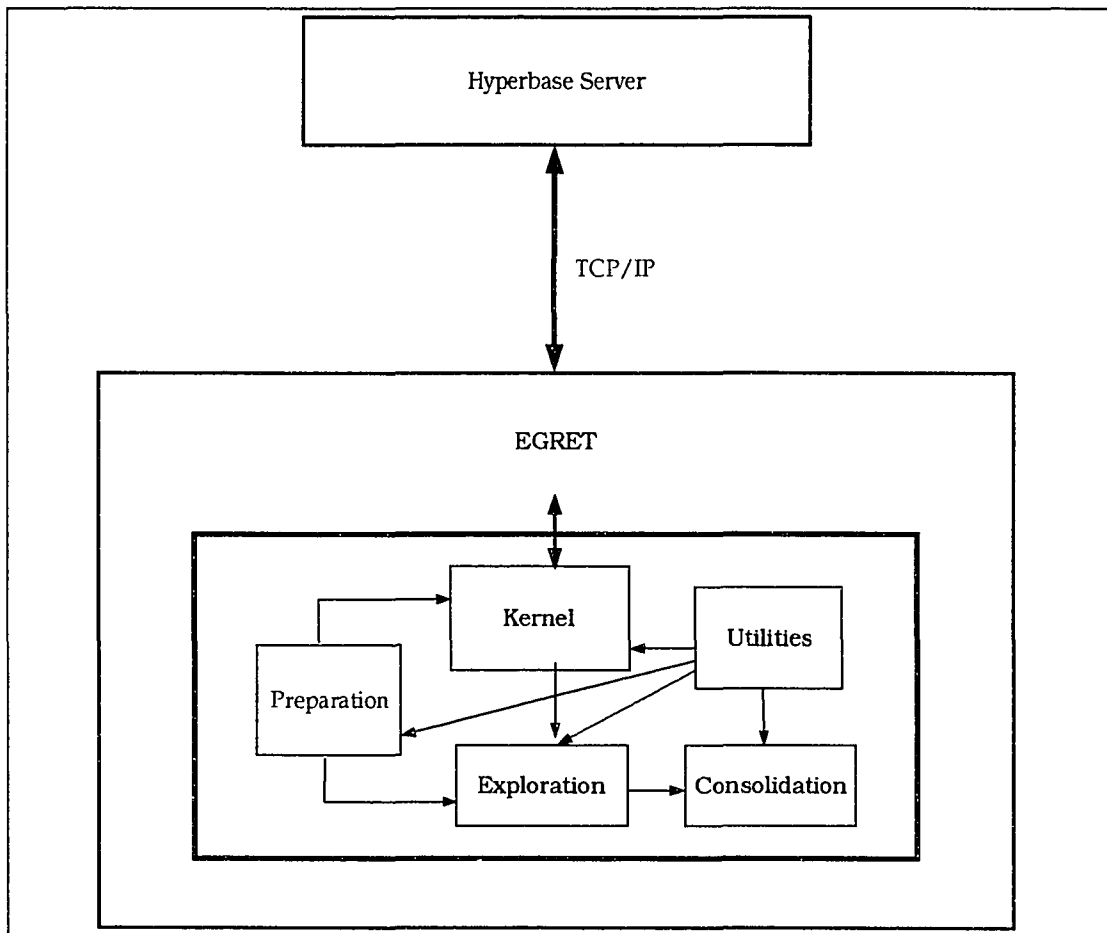


Figure 4.1: CLARE architecture

ample instances. Examples are intended to help the user clarify the semantics of RESRA primitives and guide them toward the proper use of the representation language.

4.3.2 Preparation

This module implements a set of supervisory facilities accessible only to the designated individuals, e.g., the instructor, researcher, or administrator. Its functions fall into three categories: *artifact conversion*, *session setup*, and *session tracking*. First, learning artifacts

such as research papers are imported into the CLARE database. If an artifact is available only from the printed source, it is first converted into the electronic format. The online document is then marked up in terms of *semantic units*, with each corresponding to a section or subsection. Any cross-reference, such as footnotes, bibliographic references, is also identified explicitly at this time. The marked-up document is converted into CLARE's internal hypertext format. Necessary links are added automatically by CLARE between various nodes.

The *session setup* involves defining parameters specific to a given CLARE session, which typically include participants for the current session, leading questions, CRF, and perspectives for the current artifact. The *session tracking* includes periodic checking of the status of individual participants, answering online feedback questions, and toggling phases.

4.3.3 Exploration

This module implements functions that constitute the *exploration* component of the SE-CAI learning model. It consists of two parts: *summarization* and *evaluation*. The former includes the creation, updating, and navigation of RESRA summarative node and link instances; consultation of the predefined CRFs for guidance on what to do next, viewing examples selected from previous sessions, and so on. The evaluation includes the creation, updating, and navigation of evaluative nodes, and answering the leading questions defined for the current artifact. At the invocation level, the former is very similar to that of summarization. Leading questions are a set of standard questions pre-prepared by the instructor or researcher for the current artifact. They are much less structured than the CRFs, and are used to address global aspects of different types of artifacts, e.g., major contributions.

One main feature of the exploration module is that most of its functions operate on data that is private to individual users. For example, User A is not allowed to read a RESRA

node instance created by User B. Similarly, User B cannot follow a RESRA link generated by User A. However, both users are allowed to read each other's online feedback nodes and, follow-up them as necessary. CLARE provides an access control mechanism that supports this feature.

4.3.4 Consolidation

This module implements functions that constitute the *consolidation* component of the CLARE learning model. Specifically, it consists of three parts: *comparison*, *argumentation*, and *integration*. The purpose of comparison is to highlight the differences and similarities between different users' representations of the selected artifact. CLARE supports four types of comparison: *summarization*, *template*, *evaluation*, and *leading questions*. It allows the user to easily toggle between different types of comparison, and to move back and forth between one RESRA type to another (e.g., ,) , and one leading question and another.

CLARE's argumentative functions are similar to their summarative and evaluative counterparts in the exploration module. There are two main differences, however. First, while the summarative and evaluative functions are applied primarily to the selected artifact itself, the argumentative functions focus on the user's representation and evaluation of the artifact. Second, summarization and evaluation are private activities, argumentation, on the other hand, is a direct form of *discussion* among the participants. This public nature of argumentation requires CLARE to maintain a synchronized view of the shared database.

One key feature of CLARE's support for argumentation is context-sensitivity, namely, the actions available to a user at a given point is constrained by the type of the current node. Such constraints are reflected in the options available from the popup menu.

CLARE's integrative functions are divided into three categories: (1) adding integrative links between existing nodes; (2) identifying perspectives related to the current discussion,

and associate existing nodes to those perspectives; and (3) endorsing existing positions and relationships between those positions. The user can also find out the current state of consensus among users by invoking consensus report functions.

4.3.5 Utilities

The *utility* module consists of two types of functions: generic, infrastructure-level utilities not supplied by EGRET, and miscellaneous functions that do not fit to other modules of CLARE. The former, which constitutes the majority of this module, include support for regions, screens, modes, menus, styles, and so on. Most of these functions are application independent; they are implemented in a way such that, ultimately, they might be smoothly incorporated into EGRET.

The second group includes reporting facilities, which parse the CLARE database content to produce structured, typeset hardcopy summary of session activities, and CLARE-specific instrumentation functions, e.g., extracting metric data from the database.

4.4 Interface

CLARE is implemented on top of Lucid Emacs, an X Window based version of the GNU Emacs editing environment. Most of its interface facilities, such as multi-screens, pull-down and popup menus, multi-fonts, active-regions, are inherited from the underlying platform.

Figure 4.2 represents a typical user view of CLARE during the *exploration* phase. The screen consists of three windows: one occupies the entire left half of the screen, and the other two equally divide up the remaining portion of the screen. The left window is for displaying source or artifact nodes. The upper-right window is used to list a group of related RESRA instances (e.g., all unseen claim) so that the user may selectively view

their detailed content through mouse clicking. The lower-right window is used to show the detailed content of a node. All user-level commands are available from popup and pulldown menus, both of which are sensitive to such contexts as the node type, the current phase, and so forth.

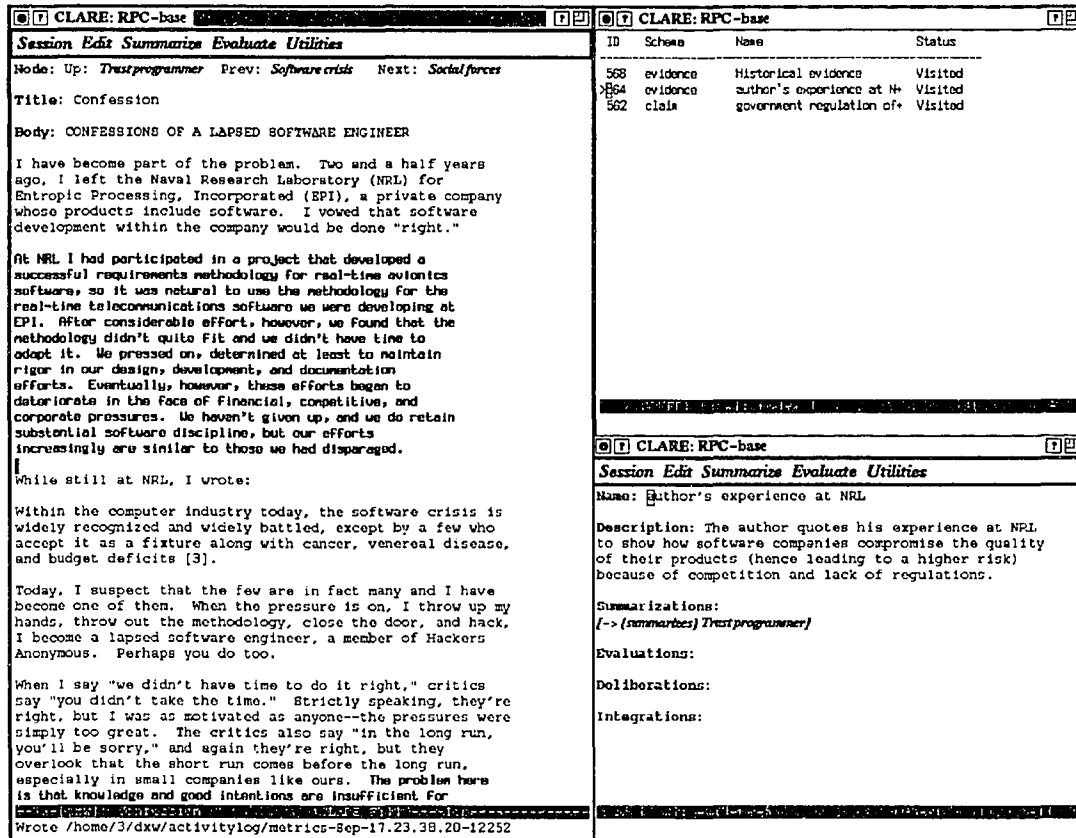


Figure 4.2: Example view of CLARE during exploration

Figure 4.3 is a typical view of CLARE during the consolidation phase. The upper left window contains a comparative view of the RESRA instances generated from the previous phase. In the example shown, it is a listing of **Problems** by Peter, Cam, and Rose — the three participants of the current session. The highlighted text (i.e., with the bold italic font) represents the links to the corresponding node that contains more detailed information, e.g., the source nodes from which the RESRA node instances are derived. In the

example shown, when the user mouse clicks the **Problem 1**, i.e., “Software discipline,” the corresponding node will be shown in the lower right window. When the user follows the link under the **Summarization**, the related source node is shown in the lower left window. The highlighted text (in bold font) is the place from which Peter derives his problem.

The upper-right window shows the summary statistics of each user has done during the exploration phase. Entries in this window are mouse-sensitive. When an entry is selected using the middle mouse button, the detailed listing of the corresponding user will be displayed. For example, if the first entry is chosen, the list of all the 19 nodes created by Peter will be shown in that window. The user may then select any node to view.

The screenshot shows the CLARE software interface during consolidation. It consists of several windows:

- Top-left window:** Titled "CLARE: RPC-base", it contains a menu bar with "Session Edit Compare Argument Integrate Utilities". Below the menu, there are three problem descriptions:
 - Problem 1: Software discipline** (highlighted in bold): "Author implies that software engineers must be forced by an authority figure (management, government) to adapt effective SE practices. Without this, well-recognized good practices will not be implemented. Furthermore, most SE's do not inherently possess this self-discipline."
 - Problem 1: Software Crisis**: "There is a struggle to make correct programs. In addition today, we require more than just correctness. We require reliability, safety and trustworthiness"
 - Problem 2: Regulation will be hard to do**: "It is difficult to know how to effectively regulate software engineering"
 - Problem 1: Unregulated Technology**: "Humans have become increasingly dependent on many technologies. However, this dependence can be..."
- Top-right window:** Titled "CLARE: RPC-base", it displays a table of user statistics:

ID	User	PR	CL	CO	HE	EV	TH	SO	TI	OT	T+
236	Peter	1	10	2	0	1	2	3	0	0	19
240	Don	2	7	1	0	6	0	0	0	0	18
232	Rose	1	8	0	0	0	0	0	0	0	7
- Bottom-left window:** Titled "CLARE: RPC-base", it contains a menu bar with "Session Edit Summarize Evaluate Utilities". The main text reads:

hands, throw out the methodology, close the door, and hack, I become a lapsed software engineer, a member of Hackers Anonymous. Perhaps you do too.

When I say "we didn't have time to do it right," critics say "you didn't take the time." Strictly speaking, they're right, but I was as motivated as anyone--the pressures were simply too great. The critics also say "in the long run, you'll be sorry," and again they're right, but they overlook that the short run comes before the long run, especially in small companies like ours. The problem here is that knowledge and good intentions are insufficient for success. This is not just true of software. I know about nutrition and my intentions are good, but my diet isn't.

I'm told that the first labor laws in England resulted in part from the efforts of those who wanted fair labor laws but who could not compete unless all businesses were required to meet the same standards. There are parallels in today's software industry.
- Bottom-right window:** Titled "CLARE: RPC-base", it contains a menu bar with "Session Edit Compare Argument Integrate Utilities". The main text reads:

Name: Software discipline

Description: Author implies that software engineers must be forced by an authority figure (management, government) to adapt effective SE practices. Without this, well-recognized good practices will not be implemented. Furthermore, most SE's do not inherently possess this self-discipline.

Summarizations:
[-> (summarizes) Confession]

Evaluations:

Deliberations:

Integrations:

Figure 4.3: Example view of CLARE during consolidation

4.5 The road map

Figure 4.4 is an overview map of process steps a user follows in a typical CLARE session. The shaded and numbered boxes on the left represent main activity steps. The unidirectional link from *exploration* to *consolidation* indicate the sequential order of the process, i.e., once the group enters the consolidation phase, they are normally not allowed to revert to the exploration phase, nor is it permissible to have the two going on at the same time. The bidirectional links connected shaded boxes within each phase show that the associated activities might be engaged simultaneously or intermixed in any order. For example, although it is typical that *evaluation* comes after *summarization*, it is also not uncommon to see the two unfolding simultaneously, or even *evaluation* invoked before *summarization*. The boxes on the right side list major sub-activities of the corresponding entries on the left. The one-line question/statements in the middle capture the basic question to be answered by the corresponding process step.

4.6 History and status

4.6.1 Brief history

CLARE, in its various incarnations, has been in existence for almost three years. During this time, it has undergone substantial changes at both conceptual and implementation levels. Because CLARE is the first system built on top of EGRET, it is not surprising that the evolutionary path of CLARE is interleaved with that of EGRET. Figure 4.5 depicts a few important milestones in the development history of CLARE.

Plover. The origin of CLARE traces back to Plover, an 800-LOC implementation of a rudimentary hypertext system based on Info, the Emacs' online documentation browsing system. Plover uses the same internal storage format as that of Info. However, it differs

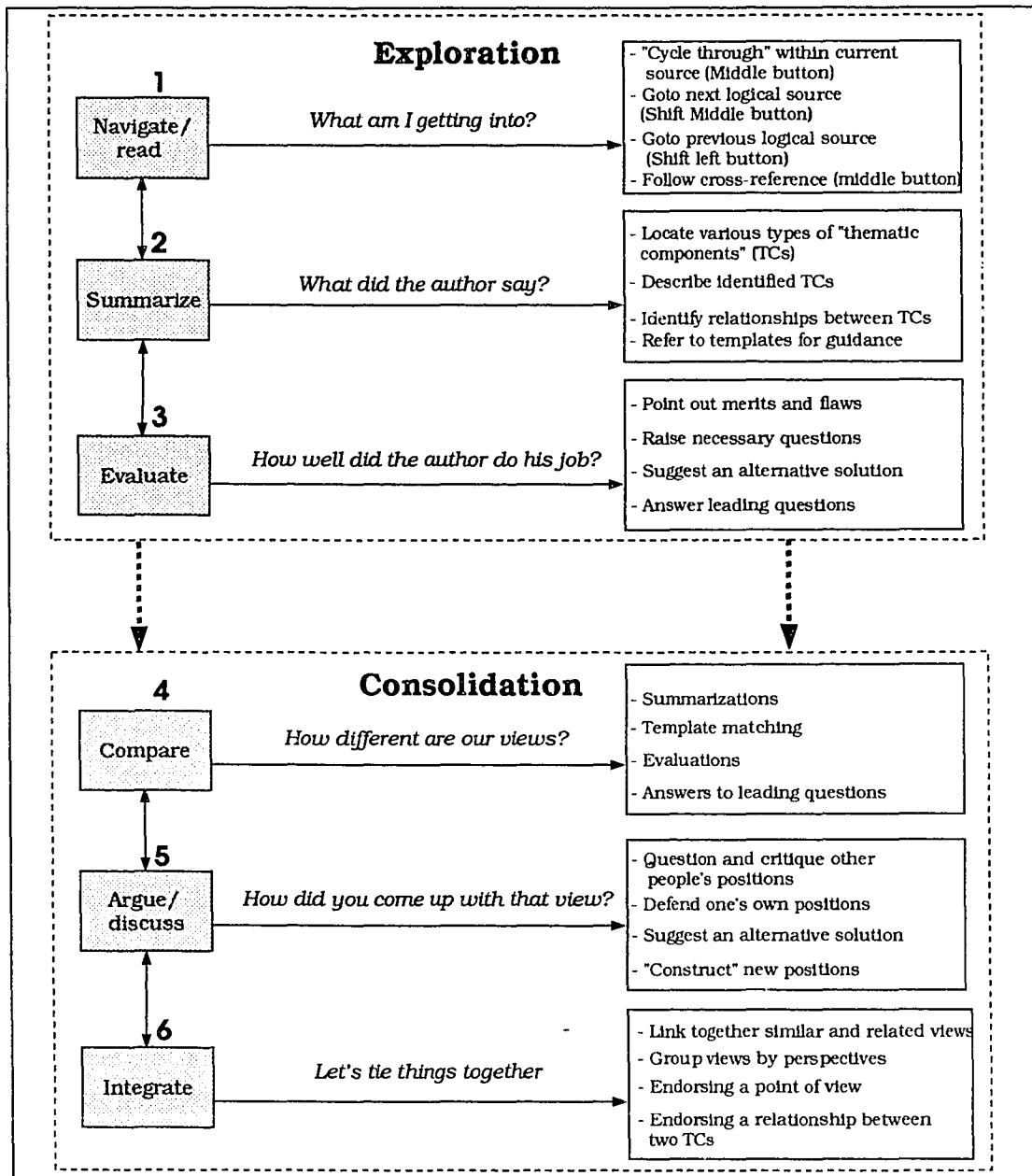


Figure 4.4: CLARE functional road map

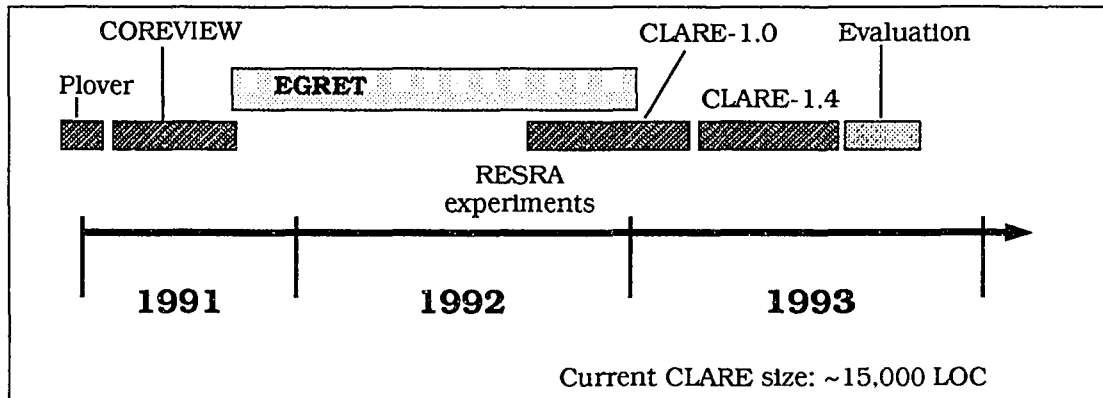


Figure 4.5: Evolutionary path of CLARE

from Info in an important way: unlike Info, which is a read-only, online browsing tool, Plover is a collaborative system that supports semi-structured group discussions about design and research. As such, it emphasizes interactive creation of nodes and links. However, as a group tool, Plover suffers from several vital flaws. First, it provides very limited distributed support. Even though the user is allowed to access and modify the shared document on the remote machine via anonymous FTP, Plover has no locking mechanism to prevent concurrent updates. Furthermore, because Plover was implemented on top of the vanilla Emacs, it does not have support for multi-screens, popup/pulldown menus, and active regions. The latter constraint forces links and field labels to be treated as plain text which are modifiable by anyone and, consequently, can lead to in corrupted data. Plover had only a brief existence; it was used by the development group a few times, and was quickly succeeded by the initial implementation of COREVIEW/EGRET.

COREVIEW: prototype and experience. Following the public availability of Hyperbase in May, 1991, a hypertext database server from the University of Aalborg, Denmark, an initial prototype of COREVIEW was implemented [WJ92]. This prototype was largely derived from EHTS, the Emacs interface to the hyperbase that comes with the distribution, which runs under Epoch, an X-window based version of Emacs from University of Illinois.

The primary purpose of this system is to support collaborative literature review in seminar settings using IBIS-like nodes and link types. In fact, COREVIEW was used, albeit briefly, in a graduate seminar on object-oriented design in the Fall, 1991, at the Information and Computer Sciences Department of the University of Hawaii. The attempt, however, was unsuccessful; frustrations and overwhelming negative feedback about the system from the students forced us to withdraw the original plan of using it in throughout the semester. The experience reveals three major problems with our initial system:

- *Fragility of COREVIEW.* The initial prototype, which was implemented from bottom-up, i.e., restructured from an existing system, rather than from top-down, i.e., based upon well-defined specifications, was not sufficiently robust and reliable for public usage. At the interface level, the lack of context-sensitive popup/pulldown menus of available commands also presented quite a barrier to the uninitialized users.
- *Lack of understanding of the problem domain.* At the conceptual level, COREVIEW provides no definite answer to such basic questions as, “what is the precise problem the system attempts to address?” “What approach it embodies?” “How does the approach addresses specific aspects of the problem?” Such ambiguities were visible from the absence of an explicit process model and a coherent representation in COREVIEW.
- *Lack of specification in experimental procedures.* Although, empirically, the initial usage of COREVIEW was guided by some preliminary research hypotheses, detailed procedures and the range of outcomes from the experiment, however, were not specified.

The above factors had led to the birth of EGRET and a complete re-conceptualization of CLARE.

CLARE and the new EGRET. Following the first unsuccessful use of COREVIEW and, for that matter, EGRET, much effort in the next 10 months was devoted to the redesign, implementation, and optimization of the platform using more rigorous development methodology, i.e., a combination of layered and object-oriented methods. A separate regression testing module was also developed for assuring the reliability of the system. Meanwhile, work on CLARE itself took place mainly at the conceptual level, i.e., characterizing the problem domain and developing a new representation-based approach to solving the problem. During the summer, 1992, six rounds of paper-and-pencil based usage experiments within the CSDL were conducted in an attempt to test the viability of the solution and refine the representation, i.e., RESRA.

As EGRET became stable and the conceptual framework for CLARE gradually came into form, efforts began to shift toward implementing the system. During next 10-months, CLARE underwent several cycles of “prototype \Rightarrow shakedown use \Rightarrow redesign/refinements.” The initial prototype, for example, does not support online browsing and marking of the full-text of the selected artifact; the user was required to read the printed artifact and create necessary RESRA instances online. The next release provides such a feature but only supports a *flat* view of the artifact, i.e., the entire artifact was treated as a single entity, which, among other things, makes it difficult to collect fine-grained process data. The subsequent prototype introduces the concept of *semantic units*, and provides functions for semi-automatically converting the linear text into a hypertext network.

4.6.2 Current status

The current implementation of CLARE consists of approximately 15 KLOC of Emacs Lisp code. Although it is still a research prototype, CLARE has been functionally stable. The system is being used on a regular basis both within the Collaborative Software Development Laboratory and in real classroom settings. The main pending change to the system is to upgrade it to use the most recent version of EGRET with improved reliability, efficiency,

and functionality. Another important extension that is currently under consideration is a graphical front-end which, based on our experience and user feedback thus far, will enhance the usability of the system by allowing direct manipulation of RESRA instances and better visualization of the relationships between various RESRA nodes.

Chapter 5

Hypotheses and Experiments

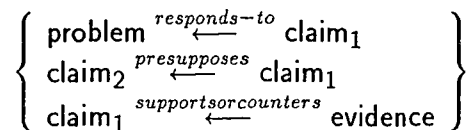
This chapter describes the CLARE evaluation procedure. Specifically, it elaborates ten empirical hypotheses, and a number of experiments designed to explore these hypotheses. Section 5.1 revisits the research problem. Section 5.2 describes each of the ten guiding hypotheses. Section 5.3 identifies and relates three types of empirical data to be collected for each hypothesis: *outcome*, *process*, and *assessment*. It also describes the procedures for gathering this data, and methods for analyzing them. The chapter concludes by discussing the actual experiments to be conducted, including the task, subject, procedure, and the execution plan.

5.1 The research problem revisited

As described in Chapter 1, CLARE addresses the problem of representation, or more specifically, the lack of representational support, in existing collaborative learning systems. It explores issues related to the use of the higher-order knowledge embedded in both scientific text as a basis of differentiating, deliberating, and integrating different points of view and interpretations from a group of learners, and of facilitating interactions among them. There are three main suppositions underlying this approach. First, scientific text, such as research papers, exhibits structural patterns that embody certain dominant norms and conventions governing the current practice in knowledge construction and presentation in a given scientific community. Furthermore, such patterns can be characterized in terms of a small number of primitives. Selected aggregates of those primitives form a core

set of structural *ideal types* of selected exemplary artifacts in that domain. And finally, the combination of those primitives and aggregates provides a useful framework for learners to collaboratively learn about the domain through the artifact-centered exploration and discussion.

For example, one type of research artifacts in software engineering research, called *experience paper*, typically contains the description of a problem situation encountered by an organization, the initial rationale for adopting a particular piece of software technology, and resulted experiential evidence on whether or not that technology turns out to be effective in that particular organizational context. In CLARE, the thematic structure of such a paper can be expressed as follows:



An example instantiation of problem, claim₁, claim₂, and evidence is as below:

- **Problem**: “Our organization has failed to produce high-quality software despite the increased staff and the state-of-the-art environment.”
- **Claim₁**: “Cleanroom engineering is the development methodology we should adopt to improve the software quality in our organization.”
- **Claim₂**: “Cleanroom engineering helps produce zero defect software using statistical quality control.”
- **Evidence**: “The average number of errors per KLOC has decreased substantially but it took us twice as long to complete a system of the same-size.”

A student in software engineering may use the above structural model as an example *template* to guide his interpretation and evaluation of other experience papers in this domain, and his efforts in constructing new experience papers so that they may also conform the same structure. In a group setting, such structural knowledge can be used as a shared framework among learners to engage in discussions about the content of related artifacts.

On the other hand, written artifacts such as published journal articles and conference papers are likely to deviate from the canonical structural protocols. For instance, a concept paper might fail to provide evidence to substantiate the viability of the proposed approach, as suggested by the standard structure. Furthermore, both structural and content-level ambiguities are commonly found in the research literature, some of which are attributable to the lack of specificity on the part of the author, while others are caused by the gap in the domain knowledge between the author and the reader. Such inconsistency and a lack of precision in the artifact itself often lead to different interpretations and controversies among its readers regarding what the artifact, or a given segment of it, really means.

The CLARE approach to collaborative learning is based on the structural characterization of the scientific text described above. It asserts that learning is knowledge construction through collaborative and critical analyses of scientific text, and through subsequent deliberation and integration of different interpretations and points of view among a group of learners. At the core of CLARE is a thematic representation called RESRA, which consists of a small number of primitives, and a set of aggregates called CRFs (Refer to Chapter 3 for detailed description of RESRA, and Chapter 4 for CLARE). It embodies a two-phase model of collaborative learning, and a set of services designed to facilitate the use of the RESRA to represent the content of selected artifacts, to deliberate the reasoning behind different points of view, and to integrate similar views into a coherent whole. See Figure 2.6 for an overview of the CLARE process model, and Figure 4.4 for a detailed function map of CLARE.

5.2 Hypotheses

The ten research hypotheses to be described below are divided into three groups: RESRA, SECAI, and CLARE. Despite their seemingly variety, most of these hypotheses are closely related, and may thus be tested using the same data sets. Table 5.1 summarizes those hypotheses and the dependencies among them.

5.2.1 RESRA

H₁

*RESRA is an effective mapping tool for characterizing essential thematic components and relationships in scientific text. The word *essential* is used here to mean the skeleton of ideas as intended by the author that form the backbone of an artifact. Certain components of the artifact may seem essential to the reader/learner but not necessarily so to the author, for example, those that coincide with the learner's interests or points of view. Such semantic gaps between the producer and the reader of a written artifact often exist and are even deemed desirable in the context of collaborative learning, for they constitute a source of potential controversy which, with proper facilitation, can lead to a better understanding of the underlying problem and discovery of new knowledge.*

H₂

RESRA is a useful organizational tool that enables learners to incrementally integrate their fine-grained interpretations of an artifact with those of their peers. The RESRA-based integration takes places at two levels, namely, integration of different interpretations by individual learners on the same artifact, and the integration of the consolidated group views on different artifacts. The focus of this hypothesis is on the former. It is important to note that integration is not a simple concatenation of two sets of related RESRA instances.

Instead, it requires careful characterization of the relationships among those instances, and grouping through aggregations, such as perspectives.

H₃

RESRA provides a useful shared frame of reference for collaborative construction of knowledge from scientific text. RESRA, in essence, is the *medium* of learning and interactions under CLARE. The content of an artifact, or individual interpretations of it, is summarized and evaluated in terms of RESRA instances. Furthermore, RESRA also provides the basic grain size for comparing different viewpoints, a framework for deliberating rationale behind those views and, ultimately, for consolidating them into a coherent corpse of knowledge. Therefore, the collaborative utility of RESRA occupies the central stage in CLARE. This hypothesis presupposes H₁, H₂, and H₆.

H₄

The Canonical RESRA Form, or CRF, provides a viable means of characterizing the thematic structure of exemplary scientific text. A corollary of this hypothesis is that RESRA can serve as a useful learning tool for the newcomers of a domain about the norms and conventions (a.k.a., Kuhnian paradigms) that govern the formal communication of knowledge via written artifacts.

H₅

The Canonical RESRA Form, or CRF, provides a useful heuristic basis for exposing ambiguities and gaps in the existing work. This hypothesis is a natural extension of the mapping capability of RESRA (i.e., H₁); the suggestive power of the CRF depends heavily on the quality of predefined CRF instances and the match between those instances and the content structure of the current artifact. Even though the CRF-based suggestions are not always

right, they nevertheless might be useful, for any discrepancy between the two suggests that either the existing CRF instance needs to be amended, or a new CRF instance should be added.

H₆

The use of CRF consultation functions can lead to a more consensual view on an artifact among a group of learners. In a sense, this hypothesis is an instantiation of H₃; more specifically, the CRF might be used to guide consensus-building among a group of learners.

5.2.2 SECAI

H₇

The dichotomy of exploration and consolidation in the SECAI model represents a viable approach to collaborative learning under the CLARE framework. This hypothesis presupposes the primacy of adequate individual preparation, which is carried out in the exploratory phase, to productive collaboration, which is accomplished in the consolidation phase.

H₈

During an extensive period of using CLARE, learners exhibit a noticeable migration from summarative activities to evaluative and argumentative activities. Such a migration may be explained by two factors: first, most learners at the beginning may lack sufficient domain knowledge. Hence, their focus is mostly on internalizing the content of the artifact rather than evaluating it. Second, since a typical group consists of a mixture of learners, some of whom are primarily summarization-oriented, while others are more critical-minded. Over time, the former may learn to lean toward the latter.

5.2.3 CLARE

H₉

The ability to make fine-grained comparison between different points of view on the same artifact is essential to collaborative deliberation and, ultimately, the integration of those views; CLARE provides such a capability. The comparative view highlights key differences and similarities of individual learners' representations at both the primitive and the artifact levels, i.e., the CRF.

H₁₀

CLARE provides a viable platform for supporting artifact-based collaborative learning. This hypothesis forms the empirical basis behind all exploratory activities using CLARE. Its formulation is non-comparative. Hence, it does not require any direct comparison between CLARE and alternative approaches.

5.3 Data collection

To evaluate CLARE and test the hypotheses described above requires three types of data: *outcome*, *process*, and *assessment*, each of which is described in turn in the following sections. The specific data for each of the ten hypotheses is also identified. Table 5.2 provides a synopsis of such data.

5.3.1 Outcome data

The *outcome* data measures the quantity and quality of what individual learners and groups have done in a CLARE session. For example, H₈ requires a comparison of the number and

Table 5.1: A synopsis of the ten hypotheses on CLARE

Hypothesis	Description	Dependencies
H ₁	RESRA is effective for representing essential content of scientific text.	None
H ₂	RESRA is effective for incremental, fine-grained integration of learners' views on an artifact.	H ₁
H ₃	RESRA provides a viable framework for collaborative construction of knowledge.	H ₁ ; H ₂ ; H ₆
H ₄	CRF provides a viable means of characterizing the thematic structure of scientific text.	H ₁
H ₅	CRF is useful in exposing ambiguities and gaps in existing artifacts.	H ₁
H ₆	CRF facilitates the generation of consensual views on scientific text.	None
H ₇	The dichotomy of exploratory and consolidation phases facilitates collaborative learning.	None
H ₈	Over time learners exhibit a migration from summarative to evaluative activities.	None
H ₉	Fine-grained comparison leads to effective deliberation and integration.	None
H ₁₀	CLARE is a viable platform to support collaborative learning.	H ₆ ; H ₃ ; H ₇ ; H ₈ ; H ₉

the content of summarative and evaluative instances generated by the same group in different CLARE sessions. While the quantitative outcome, such as the number of evaluative nodes, is precisely measurable and automatically accumulated by CLARE, the quality is not. The latter has to be assessed subjectively by the researcher in cooperation with the course instructor.

To test the ten hypotheses described in the previous session require several types of outcome measures, four of which are identified here: *collaboration*, *criticality*, *consensus*, *CRF conformance*. Two important characteristics shared by all these measures are:

- They are raw and descriptive.

- They have both quantitative and qualitative aspects.

For example, *collaboration* measures include the number of nodes created, the total size of these nodes, and the total amount of time spent by the learners during consolidation. To gain a more accurate picture about the group interaction, however, it also is necessary to examine the content of actual database to find out what is discussed and how the discussion is progressed.

Collaboration measures

The collaboration measure (COM) shows the degree of interactions among a group of learners. Quantitatively, it includes three raw metrics: the number of nodes created, the total size of those nodes, and the total amount of effective time expended¹ during consolidation. In addition, the actual nodes and links created during this phase are examined to eliminate certain spurious results, such as empty nodes, nonsense contents.

The COM represents an important measure, for the purpose of CLARE is to facilitate collaboration among human learners. Without a good measure of collaboration, rigorous evaluation of the system will not be possible. The COM is used in H₃, H₇, and H₉, and H₁₀.

¹CLARE automatically records the number of *busy minutes* a learner spends in the system. It creates a timestamp when the learner (1) invokes any nontrivial user-level CLARE commands; or (2) in the case the user has not invoked any CLARE commands within the last minute, a busy timestamp is generated if and only if the user has generated any keyboard event (e.g., moving the mouse) within the last predefined interval, which defaults to 2.5 minutes. In other words, at a given point if a learner has not done anything within the last 2.5 minutes, he is considered *idle*, and thus does not get credit for the elapsed time. On the other hand, if in the next second or two, he touches on the mouse or keyboard, a keyboard event is generated. As a result, CLARE considers him as *busy* and creates a busy timestamp when next minute point is reached.

Consensus measures

The consensus measure (CSM) indicates the degree of consensus among a group of learners in their interpretations and evaluations of the selected artifact. The measure is derived in a two-step process. First, the nodes created by each learner are grouped by their types. If two nodes by two different users are derived from the same semantic unit², the number of overlapping nodes represent a preliminary indicator of the level of consensus among the learners. Second, the actual content of all nodes are examined to make sure that they represent more or less the same information.

The importance of the CSM is twofold: first, since one primary goal of CLARE is to facilitate the generation of consensual views on an artifact among a group of learners, the CSM provides a measure of the degree to which that goal is achieved. Second, the CSM can serve as a indicator of the group heterogeneity with respect to a given problem or artifact, and thus be used as an independent variable to explain certain outcomes. The CSM is used in H₃, H₆, and H₉, and H₁₀.

Criticality measures

The criticality measure (CRM) shows the degree of criticalness a learner demonstrates in his view or interpretation of an artifact. A learner who points out two vital flaws in a particular method, for example, should have a higher CRM value than a learner who merely summarizes the author's original claims. A simple indicator of CRM is the count of the number of evaluative nodes, including critique, question, suggestion. Like the previous measures, the actual node content needs to be examined to avoid spurious interpretations.

²Semantic units (SUs) are the basic grain-size of reference in CLARE. They correspond to sections, subsections, paragraphs, or even arbitrary regions in the original artifact selected by the learner. Currently, SUs are the same as nodes, which typically correspond to sections in an artifact.

The significance of CRM is twofold: as a measure of group composition and a measure of the change in criticalness over time at both individual and group levels. The latter is used in H_3 . The former, however, can be used for assigning experimental groups, for a group with potentials for collaboration probably needs to have a right combination of critical individuals and not-so-critical ones.

CRF conformance measures

The CRF conformance measure (CCM) reveals the degree of congruence between a learner's representation of the content of an artifact and the suggested structural template by CLARE. A simple form of CCM is the count of the missing tuples returned by CLARE's CRF function. The value, however, needs to be verified with the actual node content to eliminate spurious tuples. CCM is used in H_4 and H_{10} .

5.3.2 Process data

The *process* data provides a detailed depiction of the path which leads to a particular outcome. It includes such information as the type of operations invoked, the sequence in which those operations are invoked, the instance on which the operation is operated, and the amount of time the learner spends on performing each of the operations. The exact types of process data for individual hypotheses vary widely. In H_9 , for example, appropriate process-level measures might include the frequency at which the comparison mode is entered, the average amount of time the learner spends in that mode, the type and frequency of the comparative operations invoked, and, perhaps more importantly, whether or not the visit of the comparison mode is immediately followed by some visible argumentative activities, such as the creation of new nodes. The latter, if commonly seen, allows one to infer that CLARE's comparison functions are useful in highlighting differences and triggering

controversies among learners. The process data supplements the outcome measures in that it provides detailed information which helps explain why a particular outcome has taken place. Such information is especially important when the outcome turns out to be different from what was expected.

Detailed process data is captured automatically in CLARE through its built-in, unobtrusive instrumentation mechanism. It can be output into a format recognizable by standard analysis packages. Process data corresponding to each hypothesis is described in Section 5.3.4.

5.3.3 Assessment data

The *assessment* data represents the learners' subjective assessment of their learning outcomes and processes, and aspects of CLARE, such as the utility of a given system feature, SECAI, RESRA. Such data is gathered through two questionnaires (see Appendix A and B for details) that are filled out by learners at the end of each CLARE session. Despite the subjective nature of the assessment questions, they provide important feedback on the effectiveness of CLARE in areas in which more objective measures are not available.

5.3.4 Hypotheses and data collection

H₁

To gain insights about whether RESRA is a viable mapping tool, it is not sufficient to merely look at one or more outcome indicators, i.e., CSM, or learners' subjective assessments, i.e., Questions_{1,15-16}³. Instead, it requires answering a wide range of process-level

³All numbered questions referred in this and following sections are the questions from the questionnaire in the Appendix A.

questions. For example, how do learners use RESRA to summarize and evaluate the content of a research paper using CLARE? Do they do summarization, followed by evaluation, or do they engage in the two activities simultaneously? During summarization, do learners create summarative nodes first, and then try to connect them together by adding links of appropriate type, or are they essentially *tuple-oriented*, i.e., creating one tuple at a time? What exact types of nodes and tuples are created by individual learners? Are they all different and partitioning, or are they largely overlapped with one another? Are all important elements and relationships in the artifact identified by individual learners, by a group, or by a number of groups? Are there mis-represented elements, for example, treating a problem as a claim. If so, how are they distributed? How often are the open-ended, node and link types are used? Answers to these and other related questions can be obtained by analyzing the detailed tracking data captured by CLARE, and by examining the content of the database generated by the group. It is important to note that an effective mapping tool does not imply a uniform pattern of using RESRA, nor an elimination of mis-represented nodes or links. In contrary, an effective RESRA should promote heterogeneity at the exploratory phase, which is the exact focus of the current hypothesis.

H₂

To determine whether RESRA is an effective organizational tool for integrating divergent interpretations of an artifact requires answering two sub-questions: does RESRA *promote* the variety of interpretations of an artifact in some consistent fashion? And, does RESRA allow easy *integration* of the resulted views? The two questions are of equal importance: without the former, the latter becomes unnecessary; without the latter, group collaboration does not exist. In terms of outcomes, the answers to both questions can be approximated through a good CSM indicator, measured immediately after the exploration and consolida-

tion, respectively. The learners' answers to Question_{2,4} indicate their perceived effectiveness of RESRA at the two levels. However, the most important source of insights probably comes from the analyses of the process data gathered by CLARE, since it is likely to reveal a detailed picture of how various views, if any, are derived (see the above paragraph), and how those views are eventually integrated. It will answer the key question, what exactly happened during integration? What integrative links were identified first? What was added subsequently? How did learners arrive at a consensus on a given relationship between two nodes? Was integration accompanied by argumentation, or vice versa? Did *group leaders*⁴, emerge in the consolidation process? Did the learners attempt to construct perspectives to *group* various points of view?

H₃

The hypothesis that RESRA is a useful framework for collaborative learning presumes that RESRA is an effective mapping and organizational tool (see RESRA1 and RESRA2 in Section 5.2). Hence, the process, outcome, and assessment data described above is directly applicable here. More importantly, the process data from argumentation, i.e., the process of collaboratively deliberating the outcomes from exploration, should provide insights to such questions as, how did argumentation get started at the first place? Was it triggered through comparing instances of particular types, such as leading questions or problems, or through navigating a series of related nodes? Where were *centers of controversy*: the exact semantics of a RESRA primitive, a position taken by a particular learner, or a problem identified in the artifact? Was argumentation centered on summarative or evaluative instances? What proportion of discussions was devoted to RESRA itself, e.g., what constitutes a *claim*? Or, should X be a *claim* instead of a *theory*? How was the learner's participation distributed

⁴A *group leader* is visible among the crowd because he is the person who creates most significant nodes and/or links and, more importantly, who has the strongest influence on other learners.

within the group? Were all learners able to become involved constructively regardless what they did during the exploration phase?

H₄

Whether the CRF is a viable way of characterizing the thematic structure of scientific text is in part determined by the completeness and the representativeness of the set of predefined CRF instances. The more complete and representative the set is, the stronger is the proof for the genericity of the CRF, and the more examples the learner can rely on to show them how to use the CRF. At the process level, the tracking data will reveal the pattern in which the CRF is used by the learner. Specifically, it will provide answers to questions such as, did learners start summarization with a CRF, followed by efforts to *fill-in* each blank spot by searching the content of the artifact? Or, did they create RESRA tuples first and then *retrofit* those tuples to a given CRF template? How do the learners deviate from the selected CRF in their summarization? The viability of the CRF does not dictate the answers to above questions.

H₅

The most important data source for determining whether the CRF helps exposing thematic ambiguities and gaps in an artifact comes from the Questions_{5,17}, which represent the learner's assessment of the heuristic value of the CRF. In addition, the process data can also provide evidence about this hypothesis. For example, one might infer that the CRF indeed helped the problem discovery process if a consistent pattern was found between the invocation of the CRF functions and the subsequent creation of critique and question nodes.

H₆

The degree of consensus among learners with respect to their views on the selected artifact is measurable by examining the content of summarative nodes created. If, for example, all learners have created a similar set of tuples, and corresponding nodes share the same content, the level of consensus is high. Conversely, if the similarity is only found at either the tuple level or the content level but not both, the consensus is low. The level of consensus, however, may not provide any meaningful indication of the heuristic value of the CRF if the CRF was not used by the learner. Thus, the process-level data must be interpreted along with the outcome measure. By doing so, it can answer such questions as, was the CRF actually used by the learner during summarization and, if so, what was the common pattern of usage? Was the reference to the CRF immediately followed by the creation of new summarative nodes and/or links? Was there any correlation between the usage of the CRF and the resulted consensus among the learners who were CRF users? In addition, the process data will also be useful in validating the learner's answers to Question₆. For instance, if an learner's answer to Question₆ is affirmative, but his process data reveals that he has never used the CRF consultation function, the assessment data for the current learner might be ignored.

H₇

At first glance, it seems evident that preparation should always precede collaboration, and that better preparation will result in high-quality collaboration. Is it truly so in CLARE? To test this hypothesis requires a detailed analysis of the resulted data and usage behavior at both the individual and group levels. Were *good explorers*, measured in terms of the process steps they have followed and the quality of their summarization and evaluation, are also *good collaborators*, as reflected by the frequency and quality of questions raised

on other learners' positions, explanations given on their own positions, alternatives proposed in response to what is already suggested, and relationships identified between views of different learners? Or, is it the case that there exists a *partition* of good explorers and good collaborators in a group, and that the group which has the most balanced mixture of the two tends to have the highest level of interactions, and generated the highest quality representation of the artifact? The process data will allow us to explore answers to questions such as the ones listed above. The user assessments from Questions₇₋₈ will provide a different data point for the current hypothesis.

H₈

During an extensive period of using CLARE, is there a noticeable shift of focus from summarization to evaluation at individual and/or group levels? If so, what are potential factors accounting for it? The answer to the latter in part comes from Question₉₋₁₀. The analysis of the types and the content of node and link instances created by the learner, and the process by which those instances are created will illuminate both questions. In particular, it will help answer such questions as, is there a difference in the pattern of using CLARE across experimental sessions? Is there a difference in the outcomes of those experiments? If the answers to both are positive, is there any correlation between the two?

H₉

While the presence of divergent views on a given artifact is a necessary condition for effective collaboration, the ability to discern the differences and similarities between those views plays a no less important role in triggering constructive controversies. In CLARE, this ability is realized through its comparison/contrast mechanisms, which provide four types of comparisons: *RESRA template*, *summarization*, *evaluation*, and *leading question*.

The process data will provide clues to such questions as, how is this facility used by the learner? Is there a consistent pattern of using this function, such as, learner A has spent substantially more time than Learner B and C in comparing the answers to leading questions? If yes, is the usage pattern in some way correlated with the quantity, quality, and the content of the learner's exploration, his participation in the subsequent consolidation, or his answer to Question₁₂? How often is the invocation of the comparison function followed immediately by a *burst* of new questions, criticisms, or other visible argumentative activities? Which type of comparison is most often used?

H₁₀

In terms of data collection, the current hypothesis can be viewed at two levels. First, it is an all-encompassing claim about CLARE. As such, the process, assessment, and outcome data for all previous formulations are directly applicable here. Second, since collaboration in CLARE takes place mostly in the consolidation phase, this hypothesis requires evidence to demonstrate the degree to which such activities have in fact taken place. The process data from the consolidation phase contains answers to a wide range of questions. For instance, how much and what types of interactions took place among learners in terms of the RESRA instances they have created? What are the sequences of events leading to, for example, the emergence of a new perspective or a consensus on the relationships between two points of view? How were argumentative and integrative actions intertwined within each learner and across learners? How did the consensus, if any, arrive? Are learners' answers to Questions_{13-14,18-21} consistent with quantity and quality of the RESRA instances they have created, and the process which led to the creation of those instances?

5.4 Data analyses

This primary purpose of the evaluation component of the current research is to find out how real learners use and view CLARE. The approach is exploratory. Its intent is to provide a basis on which more rigorous comparative studies might be performed. Hence, the analysis techniques to be employed in this research are restricted to primarily descriptive statistics, e.g., frequency counts. Such summary statistics on outcome and assessment data are supplemented by a substantial amount of qualitative analyses of the detailed process data and the content of the database generated by the group, guided by the questions identified in Section 5.3.4. Consistent and important usage patterns of various CLARE functions will be identified, plotted, and compared graphically. Data-based activity maps will also be drawn, which will show graphically the patterns of RESRA usages across learners and at the group level.

5.5 Experiments

The viability and the effectiveness of CLARE as an alternative collaborative learning environment, as formulated in the ten hypotheses (see Section 5.2), are tested through a series of experiments. This section describes the tasks, subjects, procedures, and the execution plan of those experiments.

5.5.1 Task

All CLARE evaluation experiments involve the same task, namely, learning about a subject domain by reviewing selected scientific text from that domain. The task consists of two components: individual reviews, followed by group deliberation and integration. Traditionally, such activities typically take place in a seminar setting, where students are as-

signed to read a common set of research papers from current journals or conference proceedings. They are asked to write reviews of those papers that not only summarize the key contributions, but also discuss major strengths and weaknesses of the author's approach, and problems or questions the student might have on the content of the artifact. The subsequent classroom discussion allows selected individuals to present their reviews of those papers. It also provides opportunity for the interaction of various points of view held by different students regarding the assigned papers. Through these activities, students are expected to gain deeper understanding of concepts, problems, methods, theories, et al, that are important to the paper and domain. At the same time, they are also expected to improve their critical skills in evaluating other people's work, identifying problems, developing alternative solutions, and working with other learners.

The experiment shares almost the same learning goal as described above. However, the procedure for realizing it is quite different (see 5.5.3). Unlike traditional seminars in which activities are either paper-based or carried out face-to-face, the experiment is conducted in the CLARE-mediated environment. All learning activities are governed by the process and data protocols defined in CLARE. For example, there is no writing of paper reviews in the original sense, nor face-to-face discussions. Instead, students study papers by creating nodes and links of selected types in CLARE. They interact with other students in a similar fashion, i.e., via reading and reacting to the online artifacts they create in the CLARE database.

The outcome from the experiment is a database containing a collection of integrated artifacts created by all students in the group, and a group of more knowledgeable learners. A linearized hardcopy of the database content can also be generated.

5.5.2 Subjects

The subjects are 16 upper-level undergraduates (i.e., juniors and seniors), who are enrolled in ICS414 (Software engineering II), and 8 graduate students who are enrolled in ICS613 (advanced software engineering). Both classes are from the Department of Information and Computer Sciences at the University of Hawaii in the Fall, 1993. These subjects are divided into groups of 4: four groups in the first class and two groups in the second class. The group stays the same through the experiment period.

5.5.3 Procedures

Each experiment involves a group of 4 students, whom are randomly assigned from the subject pool. First-time users receive a 30-minute overview of the objective, the basic approach and the overall architecture of CLARE, followed by a 30-minute demo of the basic functionality and interface features using pre-existing examples. A detailed user guide is provided to the user at this time.

Prior to the experiment, the selected research papers are input into CLARE database by the experiment coordinator in collaboration with the course instructor. All papers are broken down into a set of nodes, typically corresponding to sections in the paper. These nodes are connected together through a set of links. Additional processing, such as the conversion of explicit references into links, are also done at this time. Since graphics and figures are currently not handled by CLARE, hard-copies are provided to all subjects.

The actual experiment is divided into two phases: *exploration* and *consolidation*. The former consists of two activities: summarization and evaluation, both of which are carried out privately by each student. The purpose of summarization is to identify key elements and relationships in the paper as viewed by the student. The student is expected to use

predefined RESRA templates to guide his/her summarative activities. The key to this step is to suspend judgement. Evaluation goes beyond just critiquing of the original paper; it also involves questioning and suggesting alternatives.

The consolidation phase includes three activity types: comparison, deliberation, and integration. The comparison mode enables the student to see what the other students have done during the previous phase and, perhaps more importantly, to discern ambiguities, inconsistencies, differences, similarities in their interpretations. The deliberation process involves challenging other students' positions through request for clarifications, critiquing, identifying what is missing, suggesting additional sources or alternative views, etc. In response to the challenges from others, the student is expected to defend and elaborate his positions.

As a close-up step, integration requires students to connect together nodes and links created by different students through such actions as declaring two nodes as similar, subsuming, sharing-the-same-perspective, or related. The outcome from this process is a more complete, consistent, and perhaps consensual view on a selected artifact.

After finishing the exploration and consolidation, all subjects are asked to fill out two questionnaires: assessment and feedback. This step is followed by an informal discussion on learners' experience and problems encountered in the CLARE session just completed.

5.5.4 Execution Plan

The experiment is conducted between September and early October, 1993. There are five batches of experiments: three for ICS414 and two for ICS613. In ICS414, each batch consists of 4 concurrent experiment groups, which results in a total of 12 experiments. All concurrent sessions use the same research papers and operate under the identical ex-

perimental conditions. In ICS613, each batch consists of two concurrent sessions, which results in to a total of four experiments.

All CLARE experimental activities is carried out in the asynchronous mode. Subjects are told that they are not supposed to discuss the papers outside CLARE. The average length of the experiment is one week, which is about equally divided for exploration and consolidation. Pilot tests are conducted prior to the actual experiments to validate the questionnaires.

Table 5.2: A synopsis of data to be gathered

Hypothesis	Outcome	Process	Assessment
H ₁	CSM; frequency of RESRA instances of the type <input type="checkbox"/> other .	Action chains leading to RESRA instance creation.	Questions _{1,15-16}
H ₂	CSMs for pre- and post- integration.	Deliberation action chains leading to the integration of points of view.	Question _{2,4}
H ₃	COM; Rating on the quality of online discussions.	Network of action sequences during interaction of various points of view.	Questions ₂₋₄
H ₄	Quality of existing CRFs; CCM.	None.	Question ₅
H ₅	CCM; CSM.	Usage patterns of the Consult CRF function.	Questions _{5,17}
H ₆	CSM.	Usage patterns of the Consult CRF function.	Question ₆
H ₇	COM. Rating on the quality of online discussions.	Usage patterns in both exploratory and consolidation phases.	Questions ₇₋₈
H ₈	CRM.	Usage patterns in both exploratory and/or consolidation phases.	Questions ₉₋₁₀
H ₉	COM.	Usage pattern of the comparison mode.	Question ₁₂ .
H ₁₀	COM. CSM.	Usage patterns in both exploratory and consolidation phases.	Questions _{13-14,18-21}

Chapter 6

Evaluation Results

This chapter presents the results from the use of CLARE. Section 6.1 summarizes the experiments that have been conducted. Section 6.2 provides an overview of the results. The subsequent five sections discuss the actual results. Section 6.3 describes the findings with respect to each of the ten hypotheses identified in the previous chapter. Section 6.4 discusses main issues that arose from the use of the RESRA language, including a list of common representation errors derived from the usage data. Section 6.5 identifies several usage strategies employed by learners during summarization and evaluation. Section 6.6 presents a detailed analysis of one CLARE session. The purpose of this section is to bring together all previous discussions through a single, actual example, and to compare this example with the hypothetical usage scenario described in Section 1.1. This chapter closes with a summary of the major conclusions from this experimental evaluation of CLARE.

6.1 Summary of the experiments

Five sets of CLARE experiments were conducted between September 2, 1993 and October 12, 1993. The first three sets involve 16 students from a senior-level, second-semester software engineering class. This class was divided into 4 groups, which correspond to the 4 pre-existing project groups. The experiment was repeated three times on different research papers, which resulted in a total of 12 experiments. The subject studied by these

groups was “software quality assurance.” The following three papers were used in these experiments:

- “No silver bullet: essence and accidents in software engineering” by Frederick Brooks;
- “Design and code inspections to reduce errors in program development” by Michael Fagan; and
- “An empirical study of the reliability of UNIX utilities” by Barton Miller, Lars Fredriksen, and Bryan So.

The remaining sets of experiments involved 8 graduate students from a class in advanced software engineering. The class was randomly assigned to two groups, each consisting of four students. Each group participated in two CLARE sessions, which resulted in 4 experiments. The subject of study was “requirements engineering.” The following two papers from the pre-assigned reading list were used in the experiments:

- “The Automated Requirements Traceability System (ARTS): an experience of eight years” by R.F. Flynn and M. Dorfman; and
- “Supporting systems development by capturing deliberations during requirements engineering” by B. Ramesh and Vasant Dhar.

The duration of most experiment sessions was one week. Some sessions lasted two to three days longer because of interruptions from other class activities.

Table 6.1 provides the summary statistics of the 16 CLARE sessions. These experiments have amounted to a total of nearly 300 hours of usage time¹, and generated about

¹The usage time reported here is actually *busy minutes* the learners spent in CLARE. The actual connection time was significantly larger. See Section 5.3 for the definition of *busy minutes*.

1,800 nodes with a total text size of nearly 400 kilobytes. A total of over 80,000 timestamps about the usage process were also gathered during these sections. The relatively large standard deviation in the number of connections per session, the amount of effective usage time, the number of nodes created, and the total size of these nodes indicates that the level of participation varies greatly among individual learners. A noticeable trend can be observed from the data shown in Table 6.1: as the learners progressed from the first into the second and/or third CLARE sessions, there was a substantial decline in the usage time while at the same time an increase in the level of learning activities, as reflected in the number of nodes created and the total size of these nodes, for instance, in the experiments D and E. This figure seems to suggest that, at the outset, learners spent quite a portion of their time merely in getting accustomed to the system. As time went on, they were able to become more focused on the task on hand.

Table 6.1: Summary statistics on CLARE experiments

Experiments	No. Connections	Usage Time (hrs)	Node Counts	Text Size (Kb)
A. (4x4)	120	82.85	472	90.02
B. (4x4)	115	67.90	513	107.97
C. (4x4)	84	53.68	440	105.16
D. (2x4)	85	54.42	162	39.42
E. (2x4)	53	37.55	207	49.67
Total	457	296.40	1794	392.24
Indiv. Avg.	7	4.63	28	6.13
Indiv. Std.	6	3.19	16	4.58

Figure 6.1 shows a more detailed view of the distribution of the usage time per session spent by individual learners and groups. Despite the relatively small sample size (group $n = 6$), several patterns of participation already become visible. For example, group 3 is the most balanced but with relatively moderate traffic; group 1 and 6 are both active groups

except that the latter does not have an inactive member like learner 4 in group 1; and Group 5 and Group 1 share a similar pattern but differ in the level of participation. The average time devoted to the two phases of CLARE seems quite constant across all groups: learners spent on an average about 70% of their time in Phase I (exploration) and 30% in Phase II (consolidation). This time pattern provides a basis for explaining certain usage results to be discussed later in the chapter.

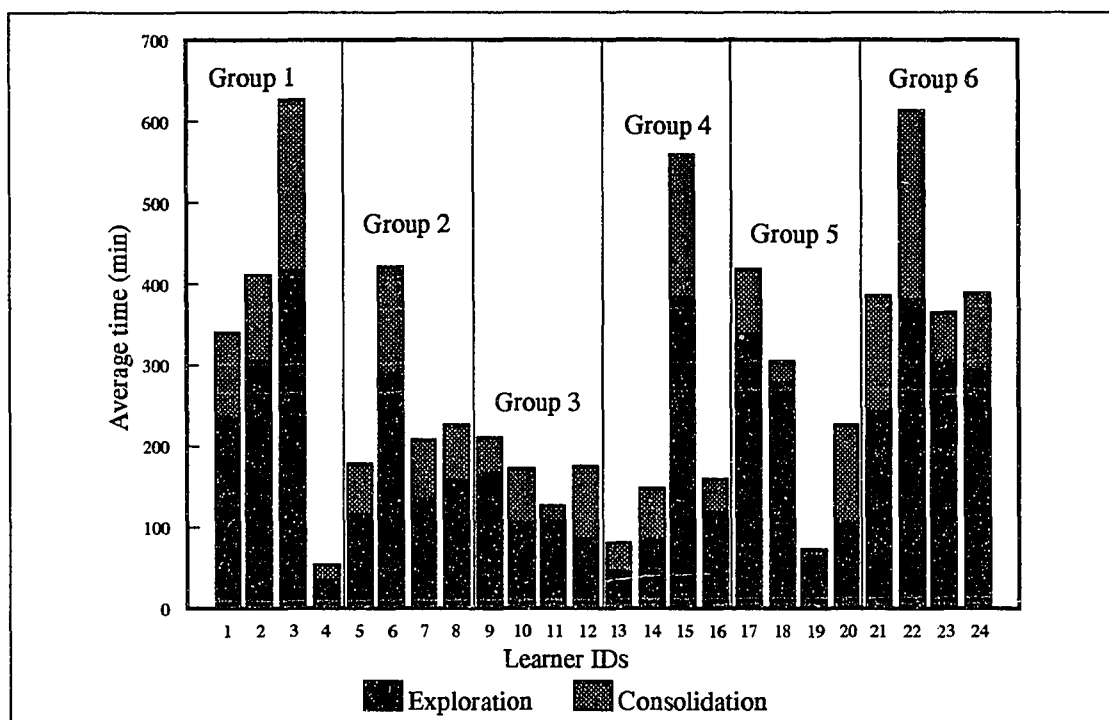


Figure 6.1: Distribution of average CLARE usage time during exploration and consolidation

6.2 Overview of the results

The CLARE experiments generated three types of data: assessment, outcome, and process. The combination of this data provides these important findings:

- Evidence about the viability of RESRA and CLARE;
- Insights about collaborative learning from scientific text using CLARE; and
- Evidence about the limitations of RESRA and CLARE.

The main findings in these three areas are summarized in Table 6.2.

Table 6.2: Summary of major CLARE experimental findings

No.	Description	Sections
RESRA ₁ .	RESRA is effective in exposing different points of view on scientific text.	6.3.1
RESRA ₂ .	RESRA is a useful means for characterizing the content of scientific text.	6.3.1
RESRA ₃ .	RESRA can be interpreted in many different ways.	6.4, 6.6
RESRA ₄ .	Effective use of RESRA requires support for representation-level deliberation.	6.4, 6.6, 6.7
RESRA ₄ .	Most learners showed difficulties in using RESRA to map the content of research papers.	6.3.1, 6.4, 6.6
CSCL ₁ .	Summarization is the most important, difficult, and labor-intensive step in the SECAI model.	6.6
CSCL ₂ .	The major themes of a research paper were not always recovered, even by a group of learners	6.6
CSCL ₃ .	Summarization alone might be sufficient as a basis for argumentation.	6.6
CSCL ₄ .	Learners adopted four general strategies in their summarization of research artifacts.	6.5.2
CSCL ₅ .	A <i>linear, one-pass</i> summarization strategy does not seem sufficient for reconstructing the major thematic map of a research paper.	6.6
CLARE ₁ .	CLARE is a novel and useful environment for supporting <i>meaning-oriented</i> collaborative learning.	6.3.3
CLARE ₂	Two most useful features of CLARE is the node primitives and the SECAI process.	6.3.3
CLARE ₃ .	The greatest barrier to using CLARE is the user interface.	6.3.3

The assessment survey and qualitative feedback from the CLARE users show that CLARE is a novel and useful learning tool: about 70% of learners indicated CLARE helped them understand the content of research papers in a way not possible before, and almost 80% of learners indicated that CLARE helped them understand their peers' perspectives in a way not possible before. The essence of CLARE, as discovered by one learner, was that "...Before I used CLARE I just read the artifacts. Now using CLARE I look for the meaning of the artifact..." Similar results were also reported on RESRA: 84% of the learners found that RESRA provides a useful means for characterizing the important content of research papers, and 90% of the learners agreed that RESRA helped expose different points of view on an artifact.

The experimental data also provides important insights on computer supported collaborative learning from scientific text. First, using RESRA to summarize the content of a research paper is the most time-consuming and difficult step in the SECAI process: most of the exploration time, which amounts to about 66% of the total usage time, was devoted to summarization. Yet, the analysis of the CLARE database reveals a great deal of misrepresentations, missed major themes, and/or links between links these themes, and so forth. Second, data from the consolidation phase shows that most argumentation was directed at summarative rather than evaluative ones. This finding implies that summarization itself might suffice as a basis for argumentation. Third, the analysis of the database content also reveals that learners often failed to identify and represent major themes of selected artifact at both the individual and group levels. Instead, they singled out many relatively minor features. And finally, the process data reveals that most learners adopted a *linear, one-pass* strategy during summarization. This finding, together with the large number of missed major themes and the links between these themes, seems to suggest that this summarization strategy is not sufficient for discerning the major themes of an artifact and the relationships between them, much less for representing them in RESRA.

The CLARE evaluation also reveals a number of problems about RESRA and CLARE. The analysis of the CLARE database shows that most learners encountered difficulties in using RESRA primitives to map the content of the research artifacts, as manifested by the large number of mis-representations. The wide variations of representation among learners and the different types of errors they committed also indicate that RESRA is subject to many interpretations. At the CLARE level, several problems also became evident during the evaluation. On the top of this list is the less-than-intuitive interface. Other factors, such as lack of online tutorial, difficulties in link creation, lack of sufficient training, and system crashes, were also mentioned.

The remainder of this chapter provides more detailed discussion on these results.

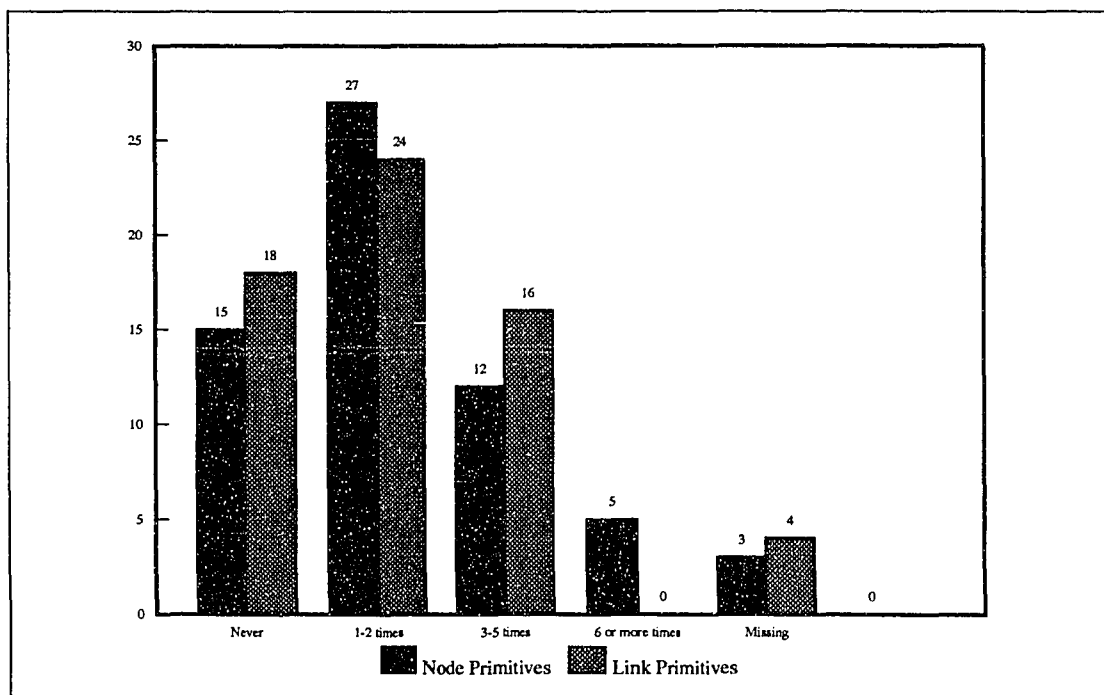


Figure 6.2: How many times did you find important contents and relationships cannot be expressed in RESRA node and link primitives, respectively?

6.3 Hypothesis evaluation

6.3.1 RESRA

H₁: Effectiveness of RESRA in representing important contents of scientific text

The post-session survey showed that 84% of users² found that RESRA provides a useful means for characterizing what is important in a research paper. Figure 6.2 also indicates that nearly every 3 out of 4 learners encountered problems with RESRA node and link primitives while using CLARE. The latter raises some deeper issues related to the interpretation and use of RESRA. A close review of the content of the CLARE database reveals a wide variety in the level of understanding of RESRA primitives and how they should be used. For example, some learners showed a good grasp of RESRA by their correct use of the node and link primitives. On the other hand, cases were found in which about half of the nodes created by the learner were used incorrectly. Such a high error-rate indicates that, in those cases, the learner lacks a basic understanding of RESRA, the content of a paper, or both. Most people showed at least one representation error during the experiment. The error rate within each individual also tends to vary greatly from paper to paper. No apparent correlation, however, was found between the type of papers studied and the rate of representation errors.

The outcome data indicates that important thematic components of a paper were not always represented, sometimes even at the group level. Instead, many minor themes were represented. In addition, relationships between these thematic components were often omitted: 23% of users did not link the summarative nodes they created. Common representational differences and RESRA usage errors are discussed in Section 6.4 and 6.6.

²The assessment survey was conducted after each CLARE session. Since all learners participated in multiple sessions, they were counted as separate entries.

To conclude, RESRA was viewed as useful despite that it was poorly understood and often, incorrectly used.

H₂: Effectiveness of RESRA in facilitating the integration of different points of view

RESRA was found useful in exposing different points of view on the artifact among different learners: 90% of the users indicated that they either agree or strongly agree with the above statement. The claim is further substantiated by the variety of representations of the artifact found in the database. Although 64% of the users affirmed that CLARE supports the integration of different points of view, the actual outcome and process data indicates that integration was often not done. One of the reasons might be the current CLARE interface, which does not allow simultaneous display of many nodes on the screen, and thus does not support smooth integration.

To conclude, RESRA is effective in exposing different points of view but its facilitative role in the integration of these views is not yet evidenced.

H₃: Viability of RESRA as a framework for collaborative construction of knowledge

The confirmation of the previous two hypotheses indicates that RESRA represents a viable framework for facilitating collaborative learning. However, as shown in Section 6.4, the usefulness of RESRA as a collaborative tool was limited by the general lack of a shared understanding of the semantics of the representation language. Because different learners had different interpretations of, for example, what constitutes an evidence or method, the resulting representation became difficult to compare and integrate.

The analysis of the CLARE database indicates that most of argumentation in the consolidation phase was triggered by summarative instead of evaluative nodes. Part of the

reason might be the relatively small number of evaluative nodes created during the exploration phase: half of the users did not create any evaluative node in this phase. On the other hand, it also indicates that summarization by itself might be sufficient as a basis for collaborative deliberation. Furthermore, evidence on RESRA-triggered discussions was also found in the database. The following argumentative nodes are two examples:

- Type: critique
- Subj: Nice structure.
- Desc.: I like the way you structured your analysis. Proposing a theory and backing up different parts of the theory with different parts of the paper. I think you've effectively captured the essence of the paper. You've reached CLARE nirvana.
- Type: critique
- Desc.: Again I'm not try to "flame" you, but you've only found two evidences to support it and I found three and called it pure conjecture. I felt this way because the author failed to present any counter evidence (perhaps because he knew doing so would prove him wrong?).

H₄: Effectiveness of CRF in characterizing the artifact-level thematic structure

The survey shows that 67% of the users indicate that the predefined CRFs captured the main thematic structure of selected research papers. However, the process data also reveals that only 28% of the users actually invoked any CRF related functions, and only 18% of them invoked CRF functions multiple times. This data indicates that many learners were unaware the availability of CLARE's CRF capabilities, much less able to exploit them during summarization.

H₅: Effectiveness of CRF in helping expose ambiguities and gaps in the content of an artifact

Figure 6.3 shows the distribution of the frequency at which CRFs had sensitized the learners about ambiguities and gaps in the research paper: 31% of the users indicated that CRFs had never helped them, while 63% of them indicated that CRF had sensitized them at least once or twice about weaknesses of the paper they had just read.

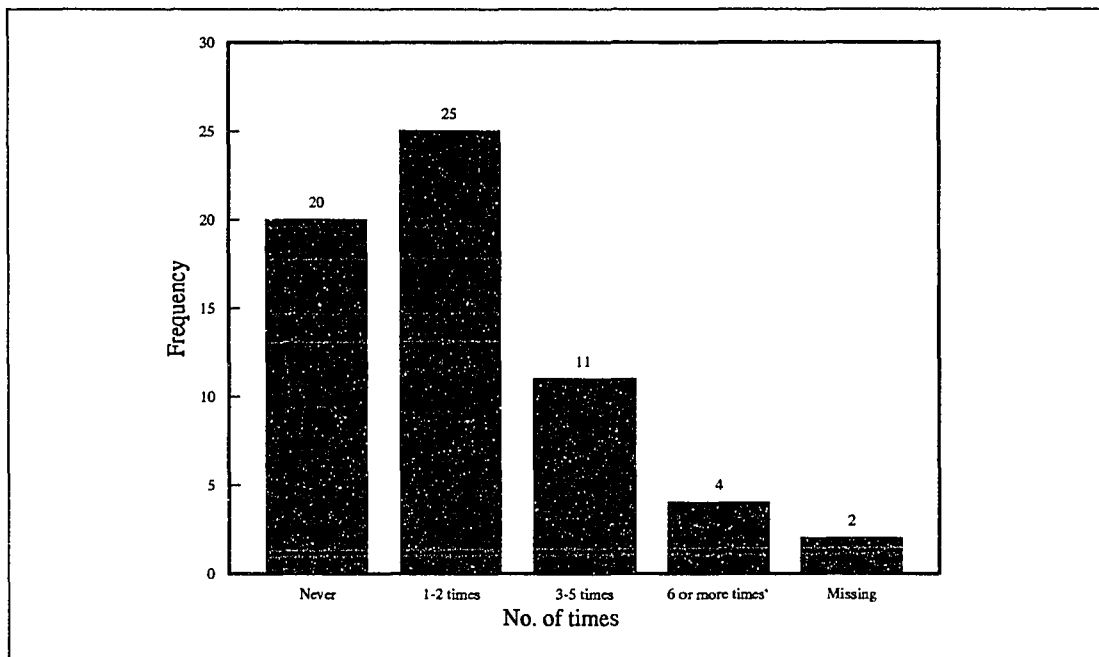


Figure 6.3: How many times did CRFs sensitize you about ambiguities and gaps in the paper?

H₆: CRF helps lead to a consensual view on a learning artifact

Evidence from the database does not seem to confirm this hypothesis. The main reason seems because CRFs are only useful for modeling the structure of major themes of an

artifact, while many learners were found representing both major themes and minor themes. In the latter case, the major themes often became *lost* in the mist of the minor ones.

6.3.2 SECAI

H₇: The dichotomy of exploratory and consolidation phases facilitates collaboration

The survey showed that 86% of the learners agreed or strongly agreed that the CLARE's two-step process model encourages each learner to do his or her part; 92% of them indicated that the model in fact facilitates the formation of individual views on the paper. However, current experiments do not provide empirical data to either support or counter the claim that the SECAI process model facilitates explicit collaboration among users.

H₈: Over time learners exhibit a migration from summarative to evaluative activities

Although 61% of the users agreed that the ability to see their peers' points of view helps improve their skills in evaluating research papers, the data from the experiment showed no such migration took place at either individual or group levels. One explanation might be that the testing of the above hypothesis requires a longer duration experiment than that performed here.

6.3.3 CLARE

H₉: Fine-grained comparison leads to effective deliberation and integration

Nearly 80% of the users found that CLARE's comparison mode is useful in highlighting different points of view. Since the current version of CLARE does not provide precise metrics on how this mode was used, it is difficult to assess whether the above subjective

view is consistent with the usage pattern, for example, whether an argumentative node was created due to the use of the comparison mode. It is evident, however, that the usefulness of the current comparison mode diminishes rapidly as the number of nodes of a given type (e.g., `claim`) increases. Since the average number of summarative nodes in the current experiment was relatively large, it might have limited the potential usefulness of this function.

H₁₀: CLARE is a viable platform to support collaborative learning

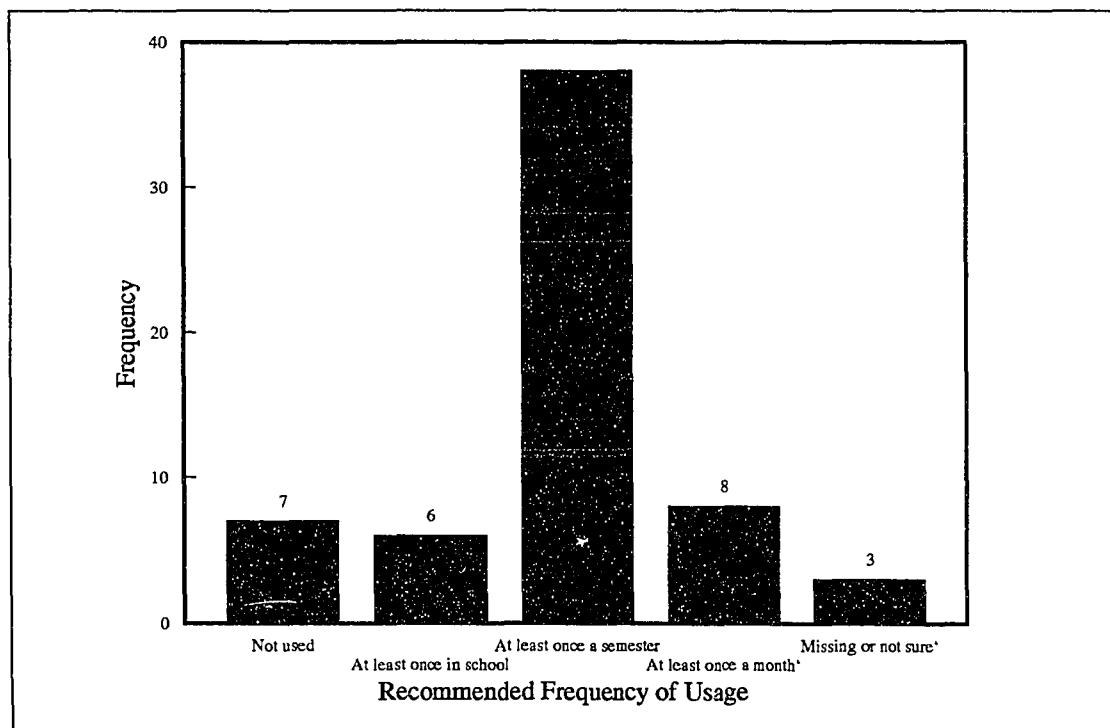


Figure 6.4: How often should CLARE be used in classroom settings?

Assessment data from CLARE users shows that the system represents a viable means for supporting collaborative learning from scientific text. About 70% of the users indicate that CLARE helped them understand the content of research papers in a way that was not

possible before. Almost 80% of the users indicate that CLARE helped them understand their peer’s perspectives in a way that was not possible before. Figures 6.4 shows further evidence on the viability of CLARE: about 70% of the users indicate that CLARE should be used in classroom settings at least once a semester. This figure is essentially consistent with Figure 6.5, which indicates that, if CLARE were currently available for public usage, 65% of the users would recommend using it to study research papers.

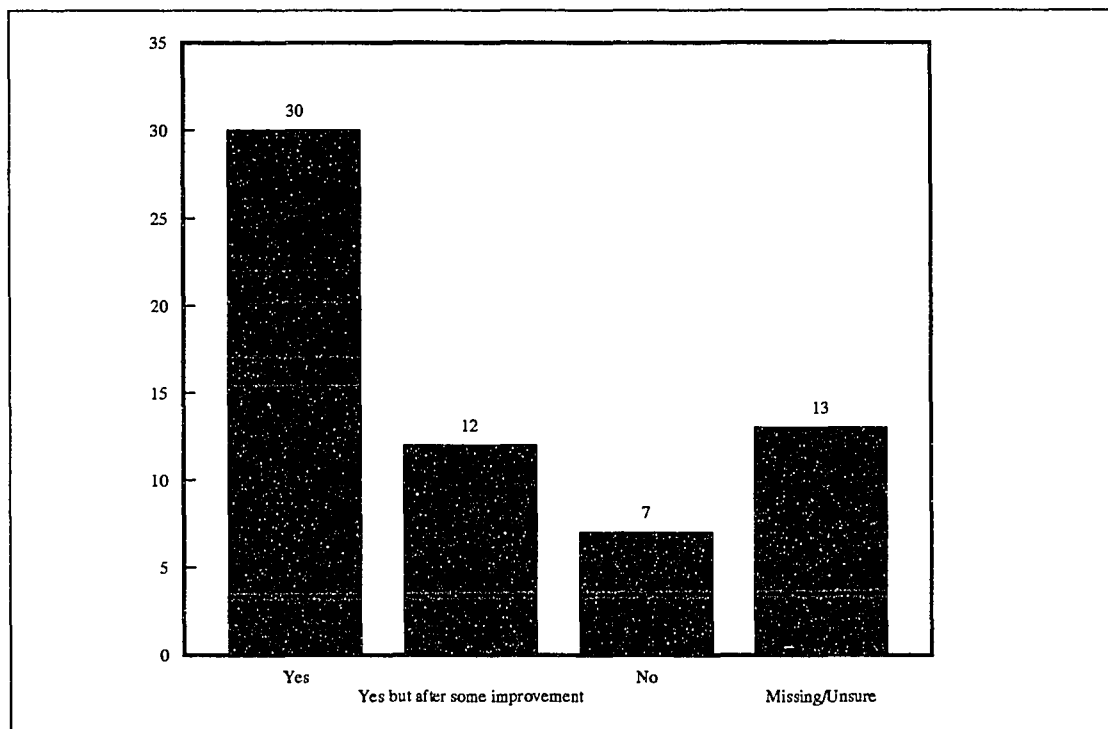


Figure 6.5: Would you recommend using CLARE for studying research papers?

In addition to the above quantitative indicators, users’ comments, such as the ones quoted below, also highlight the usefulness of CLARE:

“... I would just like to say that I really like CLARE. I don’t quite know how to use it very well yet, but it really helped me get more out of the artifact we read. Without CLARE I would have just read the artifact and not really studied it or learned about the subject. CLARE made me look at the artifact from another point of view. That

point of view was what is the author trying to tell me and how is the author trying to tell me that information. This point of view is new to me. Before I used CLARE I just read the artifacts. Now using CLARE I look for the meaning of the artifact and learn more about the subject. I would like to continue to use CLARE to read more papers.”

“I think CLARE can be used for all types of research papers, not just SE [software engineering]-related ones. There might be a few adjustments/amendments needed but CLARE already is designed to accommodate any type of change (as I see it).”

Figure 6.6 shows the relative ranking by the learners on the usefulness of CLARE user-level functions. The *RESRA node primitives* and the *SECAI learning model* were ranked the highest, viewed by 42% and 40% of the users, respectively, as *extremely useful*. At the bottom of the list was the *online examples*: 25% of the users considered them as *not useful at all*. *RESRA templates*, *comparison mode*, and *link primitives* received mixed reactions from the user. See the hypotheses H_1 , H_5 , H_9 in this section for possible explanations about the last three rankings.

Figure 6.7 shows the ranking of barriers that might have prevented users from effectively using CLARE. At the top of this list is the *user interface*: 70% of users considered it at least *somewhat an obstacle*, and 25% of users considered it as a *great obstacle*. At the bottom of this list is the *node primitives*: 47% of users did not consider it as an obstacle at all. *Link primitives* and *link mode* (the mode for creating links between nodes) were considered as moderate barriers.

In addition to the system-level barriers mentioned above, the initial usage of CLARE was also negatively affected by the following factors:

- *Time-consuming*. This factor seems already evident from Table 6.1, which shows that each learner spent on an average of almost 5 hours on each CLARE session. The same concern was also raised a number of times during feedback. The following user comments echo this sentiment:

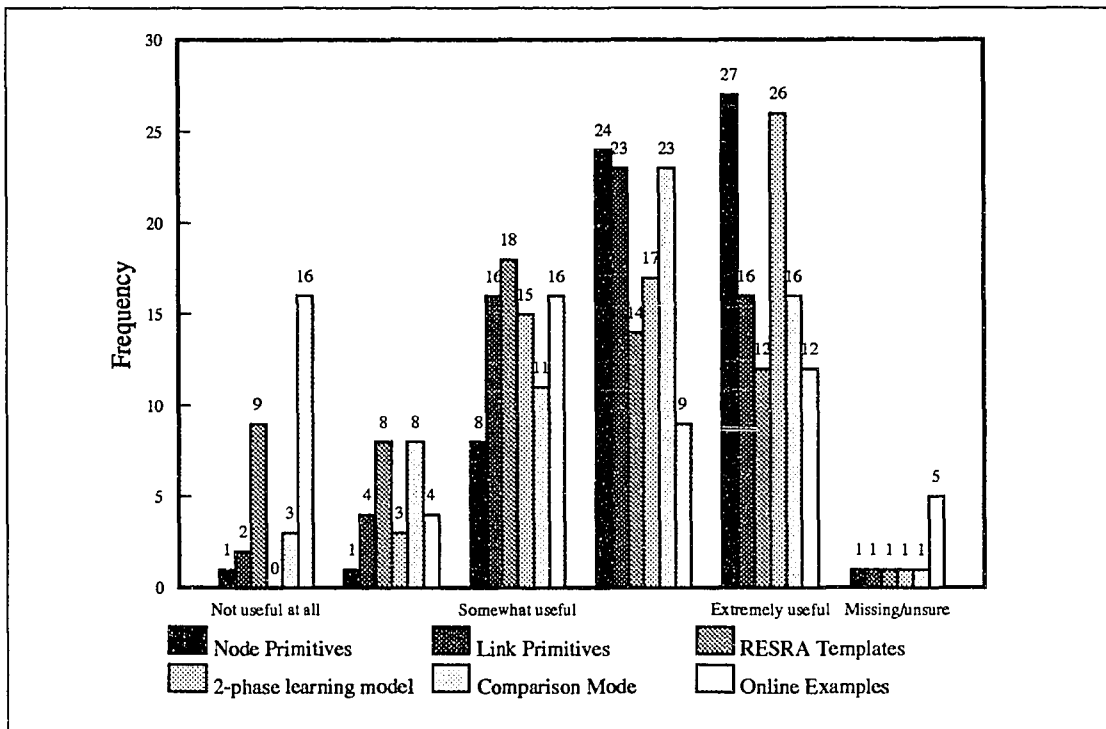


Figure 6.6: Learners' ranking of the usefulness of CLARE functions

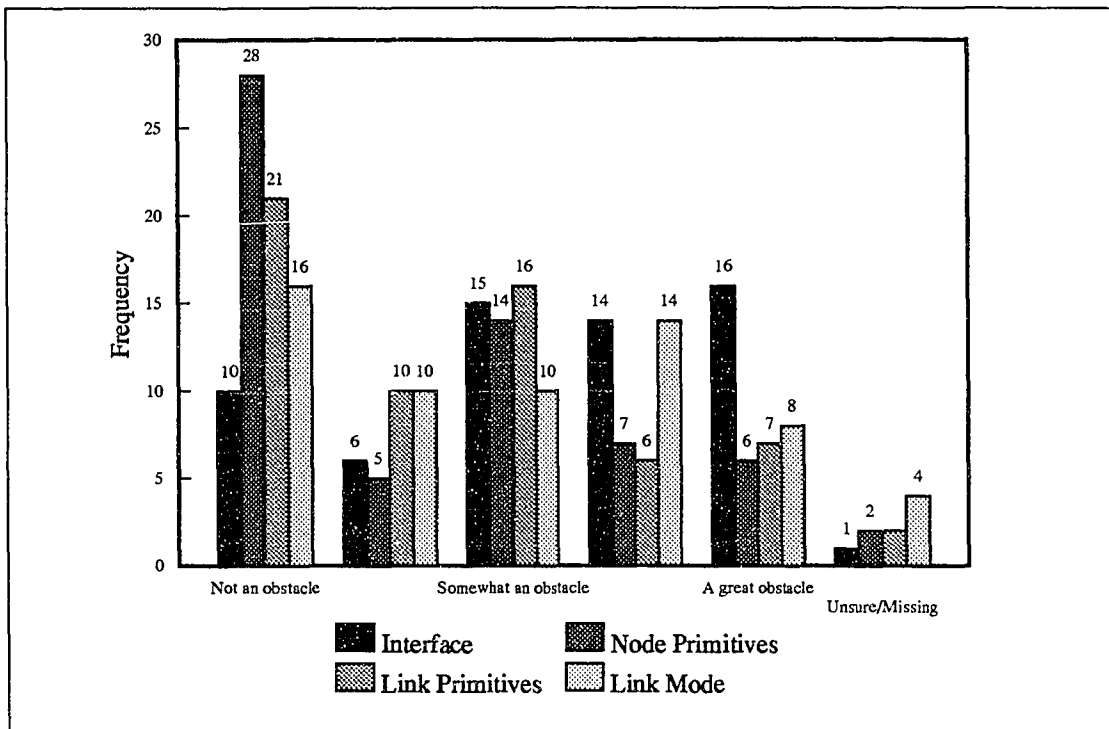


Figure 6.7: Barriers to effective use of CLARE

“CLARE is very time consuming. The time it takes to evaluate a paper with CLARE is lengthier than the time it takes to write a critical evaluation the old fashion way.”

“I think that it [CLARE] is very interesting. Unfortunately I haven’t been able to give my complete effort on this project because I am already spending 20+ hours a week on other things from ICS414.”

“I think CLARE is very useful for papers that are worth spending a lot of time on. We can’t afford, though, to spend so much time on a paper.”

- *Lack of adequate training.* As part of the experimental procedure, an hour-long overview and demo of CLARE were given to all new users. However, such a level of training does not seem sufficient. In fact, one user mentioned during a post-session discussion that at least one or two class-sessions should be devoted exclusively to teach students how to use RESRA. The situation was further complicated by the lack of online help and tutorial in the current version of CLARE, and a relatively tight schedule within which the experiments were conducted. The following user remark says it all:

“As with all powerful system[s], it takes a while to learn. I believe the students frustration will be less as they get more familiar with the package. A tutorial (online) might help.”

- *Improper selection of the research paper.* All papers used in the CLARE experiment are research-oriented. To undergraduate learners such as those from ICS414, who were not exposed to the research literature before, the content, style, and even the length of such text might be a source of problems, as shown in the following comment:

“We should have worked with a small article so that we had time to actually test the function in CLARE ...”

Despite the above system-level and environment barriers, CLARE was found to be a novel and useful tool among its users. One key source of its novelty and usefulness comes from the representation-based paradigm to collaborative learning. The following section

discusses some common representation problems learners encountered during their use of the system.

6.4 Issues on the RESRA representation

The analysis of the CLARE database has revealed (1) a wide variation among different learners in what contents they chose to represent, and how they represented them; and (2) a wide variety of incorrect usage of the RESRA representation. This section uses selected examples from the experiments to illustrate some main differences in the usage of the representation, and to describe a list of common errors in using the RESRA primitives.

For a in-depth view of these variations, see Section 6.6, which provides a detailed case analysis of a CLARE session.

6.4.1 Categorizing representational differences

The outcome data suggests that the following factors might have contributed to the wide variations of the representations among users: the granularity of representation, the distinctions between major themes and minor themes, between learners' views and authors' views, and between connotative and denotative interpretations of RESRA primitives.

“Major themes” versus “minor themes.” RESRA was designed to represent essential themes of scientific text, and for evaluating, deliberating, and integrating these themes. In other words, its focus is on the *major themes* rather than the *minor themes* of the artifact. Figure 6.8 provides an example RESRA representation of [Fag76], which consists of 11 nodes that describe the major themes of that paper.

Table 6.3 shows a comparison of the summarization result of [Fag76] by 16 learners

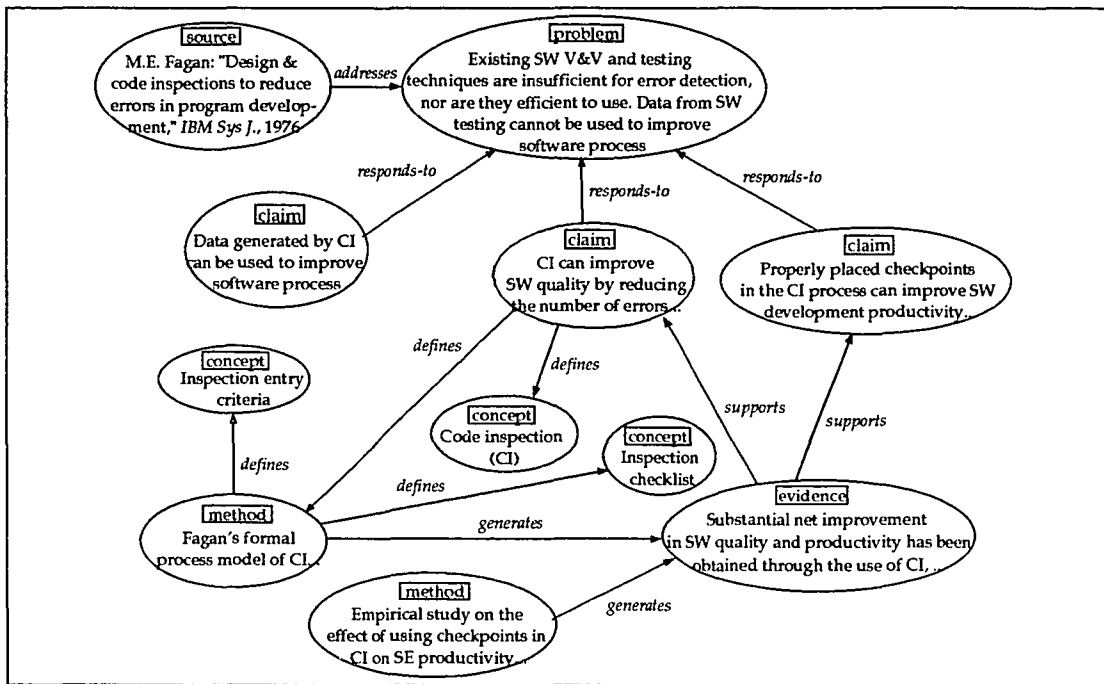


Figure 6.8: A RESRA representation of major themes of [Fagan76]

Table 6.3: Comparison of 16 learner's summarizations of [Fagan76] with the *base representation*

User	problem	claim	concept	method	evidence	theory	thing	other	Total
1	2	10	1	3	5	1	4	0	26
2	1	2	0	1	3	0	0	0	7
3	2	6	3	4	0	1	0	0	16
4	1	8	3	5	10	0	9	0	36
5	1	4	0	3	1	1	0	0	10
6	0	4	0	4	8	0	0	1	17
7	2	7	0	2	3	1	0	0	15
8	1	8	0	8	11	0	0	0	28
9	0	11	0	0	1	1	0	0	13
10	0	8	3	5	1	2	0	0	19
11	0	6	1	1	3	1	0	0	12
12	2	27	1	2	6	3	0	0	41
13	1	11	0	3	6	1	0	0	22
14	0	8	0	2	6	0	0	0	16
15	2	13	2	1	3	0	0	0	21
16	1	4	1	1	2	0	0	0	9
Base	1	3	3	2	1	0	0	0	10

and the *base representation* shown in Figure 6.8. Given the large average number of nodes created (average = 19) per learner, one would expect that virtually all the major themes of the paper be captured. The analysis of the content of those nodes shows a different picture: none of the 16 learners had the right problem; only 7 learners were able to correctly identify one or two of the three major claims; 10 learners had the evidence right; and 6 learners had one of the two methods right. These figures indicate that most attention of these learners was devoted to minor themes instead of major ones.

The current version of RESRA/CLARE does not differentiate major themes from minor ones, which perhaps was partially responsible for the problem. On the other hand, this differentiation might have been of no help if the problem learners had was not that they did not know how to use CLARE to represent major themes but that they could not have identified those major themes even if they had tried. The latter is exactly a learning deficiency that CLARE attempts to help the learner overcome.

Granularity. Another source of representational variation is the *grain size* used by learners in the node instances they created: some learners tend to create large nodes, which bring together, for instance, all related evidence with the corresponding claim, or vice versa. Others tend to generate relatively small nodes, which sometimes split a single theme into multiple nodes. For example, the following claim describes 6 reasons that justify the authors' approach:

- Type: claim
- Subj: 6 reasons
- Desc.: There are six reasons stated by the author why study of testing is important:
1. To contribute testing community a list of real bugs, which help the researchers to evaluate testing cases and verification strategies. 2. We have found the bugs that provides the security holes, therefore, additional bugs might be able to predict future security holes. 3. Some errors are caused by careless inputs. Some unexpected errors might be able to recover through this method. 4. We would like to

have some meaningful and predictable response. 5. Noisy phone line shouldn't crashed the system. 6. Lastly, we would like to compare between this new method with the traditional testing strategies.

For the same content, however, another learner created 6 separate **claim** nodes, one of which is shown below:

- Type: **claim**
- Subj: Importance of our procedure (1)
- Desc.: Provides a large list of real bugs that can provide as test cases to be used for more sophisticated testing and verification strategies.

Examples variations in granularity such as the one above are quite common in the CLARE database. This finding calls into question the usefulness of any findings based solely on the node count.

“Learners’ views” versus “authors’ views.” In CLARE, the purpose of summarization is to capture the conceptual themes of a paper as intended by the author; learner’s opinions at this stage are represented using RESRA evaluative primitives, i.e., **critique**, **question**, and **suggestion**. The analysis of the CLARE database uncovered RESRA node instances violating this rule: for example, learners sometimes state their own **problem** or **claim** as if it were the author’s. The following node was created during summarization:

- Type: **claim**
- Subj: Careless Programming
- Desc.: If return codes are not checked by the programmer, it is most likely a sign of careless programming. Time and effort was not taken by the programmer to insure that values returned by a module is indeed valid and correct and therefore causes errors.

As pointed out by another learner in the following node created during the consolidation phase, the view expressed in the above node was not the author's, but of the learner who created the node:

- Type:
- Subj: Your opinion?
- Desc.: You said that because return codes are not check that it is grounds for care-less programming. Is this your opinion? Or does the author really "claim" this? I traced your node back to the source node and did not get the same opinion after reading it.

The above confusion of the learner's view and the author's view seems partially attributable to the fact that the current CLARE does not allow the learner to use other RESRA primitives except the evaluative ones to express his or her views during summarization. For example, learners cannot make a counter or alternative claim with the author, although they can do so with each other during the consolidation phase.

"Connotative" versus "denotative" interpretations of RESRA. RESRA's vocabulary consists of a set of familiar, everyday terminologies, such as , . It redefines those terms with new and more specific meanings. However, as the CLARE usage data from the current experiments suggests, many learners still rely on the connotative meanings instead of the denotative ones when using RESRA. A large number of incorrect usages described in the next section are related to connotative interpretations of RESRA primitives.

6.4.2 Common errors in using RESRA

This section documents a number of common incorrect usages of RESRA node primitives found in the CLARE database. The purpose is twofold: to show the extent and the va-

riety of mis-interpretations of the representations by the learners, and to provide a set of examples so that future users of CLARE might avoid similar pitfalls.

Treating learner’s “problem” as the author’s “problem.” The node below is an example of treating the learner’s disagreement with the author’s position as a **problem**. The node should be a **critique** instead:

- Type: **problem**
- Subj: Flawed Werewolf Allegory
- Desc.: The werewolf allegory used to suggest the need for a silver bullet is flawed. The author argues that werewolves are terrifying because “they transform unexpectedly from the familiar into horrors.” This is false. Werewolves are indeed predictable and will only transform during the times of the month when there is a full moon.
The same predictability can be applied to the familiar software project. If the design phase is carried in a precise and detailed manner, problems associated with the project can be predicted and ultimately avoided.

Treating **evidence as **claim**.** The following node is derived from a discussion on software changeability; it is one piece of **evidence** used by the author to support the claim that changeability is an essential attribute of software systems, not a **claim** which he intends to elaborate:

- Type: **claim**
- Subj: changes due to machine vehicles
- Desc.: New computers, disks, displays are always appearing. Software must be changed to conform to these new machine vehicles.

Treating **claim as **theory**.** The following representation is an example of incorrect use of **theory** where **claim** is more appropriate:

- Type: **theory**
- Subj: No single development improves the situation
- Desc.: No single development aids in improving the software problem, at least not with respect to productivity, reliability or simplicity.

claim may not be “neutral.” A **claim** represents a *position* which can be *supported* or *countered*. The following node is *definitional* and neutral, and thus should be a **concept** instead:

- Type: **claim**
- Subj: Essence of software entities.
- Desc.: The software entity is made up of data sets, relationships among data items, algorithms, and invocations of functions. Software entities are abstract as well as precise and detailed.

concept that makes a “claim.” The following node makes a claim about a property of software systems, i.e., that software is invisible. It does not define the concept of “software system”:

- Type: **concept**
- Subj: Software is invisible
- Desc.: Software is invisible and unvisualizable. It is not easily captured by a geometric abstraction like other concepts.

A “concept” cannot be a **theory**. A concept might be a building block of a theory but itself cannot be a theory. The following node should be a **concept** instead:

- Type: **theory**
- Subj: Hawthorne Effect
- Desc.: The human bias of Hawthorne Effect supposedly plays a role in any production experiment. If not handled adequately, it isn’t clear whether the effects observed in a production experiment is due to the Hawthorne Effect or to newly implemented changes.

Treating `concept` as `thing`. A `thing` is “an object, event, or process that is the target of an inquiry.” “Silver bullet” is used metaphorically in the current paper, and should be identified as a `concept` instead:

- Type: `thing`
- Subj: silver bullet
- Desc.: something that will kill the “monster of missed schedules, blown budgets, and flawed products.”

Too “mundane” as a `concept`. In a group that is devoted to requirement engineering and a paper that exclusively describes such a system, the following `concept` node seems redundant:

- Type: Concept
- Subj: Requirement
- Desc.: An authorized and documented need of a customer.

“Explanation” ≠ `theory`. One primary purpose of a theory is to help explain a phenomena. However, the explanation itself does not constitute a theory:

- Type: `theory`
- Subj: complexity
- Desc.: Complexity is at once both the main reason behind the power of software systems, and the main source of problems in developing it.

Learner’s “claim” does not belong to `critique`. The following node contains the learner’s claim on high-level languages; it does not identify weaknesses in what the author did or said, as a `critique` should:

- Type: `critique`

- Desc.: High-level languages have been extremely helpful in improving the reliability of software by standardizing certain reusable functions that the programmer need not create from scratch.

evidence does not contain evidence. An **evidence** contains qualitative or quantitative fact intended to support or counter a given **claim**. The **evidence** node below does not provide the actual evidence; rather, it points out the corresponding **claim**:

- Type: **evidence**
- Subj: REMAP provides design decision support
- Desc.: This is evidence to show that REMAP provides mechanisms for maintaining and reasoning with dependencies among its primitives to arrive at design decisions.

“**Method** ≠ **method**. In RESRA, a **method** is defined as a “procedure or technique used for generating evidence for a particular claim.” The process for “creation of primitives” described in the node below is a **method** in the everyday sense but does not meet the definition of the RESRA **method**:

- Type: Method
- Subj: Creation of Primitives
- Desc.: The method adopted by the authors in arriving at a set of primitives for modeling design history is to start with a set of previously known standard primitives and then augment them on the basis of experiments made with expert analysts.

Treating critique as claim. The following node does not contain the author’s claim.

Rather, the learner attempted to point out what the author did:

- Type: **claim**
- Subj: ARTS - a viable tool
- Desc.: Several claims have been made on the robustness of ARTS. While some of them are facts, others do not have a well supported evidence.

suggestion contains no new proposal(s). A **suggestion** should include “ideas, recommendations, or feedbacks on how to remedy or improve an existing **problem**, **claim**, **method**, et al.” The following node does not satisfy this criterion, and should be a **critique** instead:

- Type: **suggestion**
- Desc.: Sure the author offers some examples of why he considers software the most complex thing that man has made. But I think that his automobile analogy is pretty flimsy (automobiles are very complex). I just don't think that he supported the claim enough. And of course he did try to counter it in any way.

”**Prediction**” ≠ **theory**. Although one important property of a **theory** is its ability to predict the outcome of a particular phenomena, the *prediction* itself does not constitute a theory. This node should be a **claim** whose truth value is yet to tested.

- Type: **theory**
- Subj: Effectiveness of Ada
- Desc.: Prediction that Ada will have provided training for programmers in modern software-design techniques.

Learner's questions but not author's **problem**. The purpose of summarization is to capture the *author's* view, not the *learner's*. When a learner encounters problems with the author's view, he or she creates a **question** or **critique** node. Below is a wrong way of using the **problem** primitive:

- Type: **problem**
- Subj: Some Questions
- Desc.: 1. What is the model of REMAP? 2. What is the relationship between the REMAP and IBIS? 3. What is the prototype that has been completed ?

6.4.3 Implications

The examples used above are merely a small sample of incorrect ways of using RESRA. Nevertheless, they have important implications in the future development of CLARE: while diversity in the interpretation and use of RESRA is encouraged during the exploration phase, to fully exploit the collaborative potential of the system requires two additional conditions to be met:

- A basic level of shared understanding of the semantics of the RESRA language, including the differentiation of denotative and connotative interpretations of the primitives; and
- A set of guidelines on the grain size of the representation, and the representation of major versus minor themes, and the learner's views versus the author's views.

Section 8.3 explores both issues in more detail. The next section shifts away from the current focus on the database to the findings based on the process data.

6.5 CLARE usage strategies

Over the course of the experiment, CLARE automatically accumulated over 80,000 timestamps that detail the process steps used by learners in their interactions with the system. Such process data provides an important source of information for understanding the usage behavior of the learner. This section presents findings from this data; it discusses various strategies used by the learners when they navigate the source artifact, summarize and evaluate its content, and deliberate and integrate different points of view and interpretations.

6.5.1 Navigation

Despite the hypertextual structure of the CLARE database, the process data reveals that the majority of learners still followed the traditional, linear way of navigating and understanding the text, that is, they progressed sequentially from one source node to the next. A complete linear traversal of all source nodes is called a *round*. Table 6.4 shows the distribution of the number of rounds among learners in various experimental sessions. Note that only 1-2 true *random walkers* were found. The data also shows that nearly three-fourths of learners completed their summarization and evaluation in less than two rounds. This figure implies that few of them have in fact devoted a separate round to browsing, summarization, or evaluation. Instead, they seemed to concurrently engage in all three activities. This result is somewhat surprising given that an implicit, *three-step* strategy, (browse \Rightarrow summarize \Rightarrow evaluate) is defined in the user documentation, and explicitly discussed during the CLARE training session.

Table 6.4: Distributions of the number of *rounds* on source nodes

No. of rounds	Exp. A (4x4)	Exp. B (4x4)	Exp. C (4x4)	Exp. D (2x4)	Exp. E (2x4)	Total
1 and below	3	2	5	2	3	15
1 - 2	7	8	7	5	3	30
2 and above	4	4	2	0	0	10
Random	1	1	1	1	2	6
Missing	1	1	1	0	0	3

There are three possible explanations for the above dominance of the linear navigation behavior. First, since the research papers used in the experiment were designed for a linear, non-hypertext environment, it is perhaps more comprehensible to approach them linearly, despite their hypertextual presentation in CLARE. Second, some learners were not accus-

tomed to the hypertext interface. They felt more comfortable with the traditional approach, as indicated in the following remark:

More and more, I think that my problems with CLARE stem simply from a dislike of hypertext. I am frustrated that I can't easily skim old material. Tracing links and following nodes with the mouse-middle button is (IMNSHO)³ a poor substitute for having the pieces of paper in front of you. When I wish to respond to something it is exasperating to have the node disappear as a new window opens up over it. Can these problems be addressed by current technology? I don't know. Are they critical? I guess that depends on one's viewpoint. To me, yes.

And finally, as shown in Figure 6.9, CLARE currently does not allow easy *random* access. For example, if a user wants to jump from the source node 1 to the source node 3, he or she first needs to go through the root node or source node 2⁴.

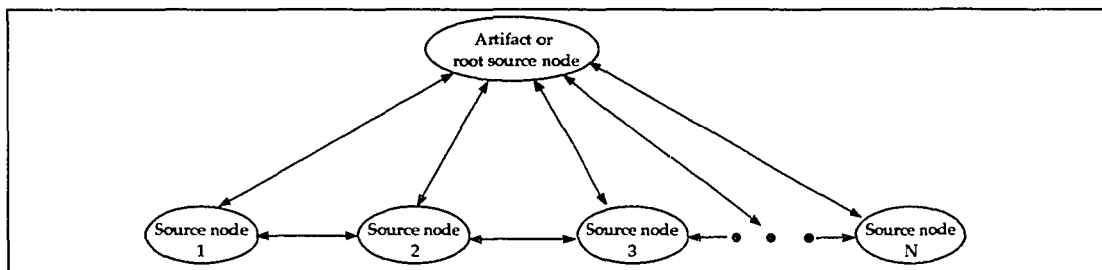


Figure 6.9: The source node structure of the CLARE database

6.5.2 Summarization

The analysis of the process data reveals that, based on the order in which nodes and links are created, four summarization strategies were present:

³Stands for "In My Not So Humble Opinion."

⁴CLARE does have an sbuff facility which allows a random selection of any source node to visit. Because of this function is available only from the pulldown menu, many users may not be aware of it.

- **Sum:** Create summarative nodes only. No attempt is made to connect them together using RESRA link primitives;
- **Sum \Rightarrow Link:** First create summarative nodes for the entire artifact. Then link them together, if possible;
- **Sum \Leftrightarrow Link:** Create summarative nodes for a single source node, or between that node and the adjacent source nodes. Then create links between them. Repeat the same process until all source nodes are summarized;
- **Sum \Rightarrow Link \Rightarrow (Sum \Leftrightarrow Link):** A combination of Strategy 2 and 3. First create summarative nodes for the entire artifact, followed by adding links between these nodes. Next, selectively create additional summarative nodes and, as necessary, add links between them.

Table 6.5 shows the distribution of the above four strategies in various CLARE sessions. Note that 36% of learner sessions adopted Strategy 1, which creates no summarative links at all. There are three potential explanations for this high percentage of absence in link creation:

- **Time pressure:** Since learners had to create summarative nodes before they could link them together, it is possible that, by the time they finish the former, they were tired, or ran out of the allocated time, or both.
- **Difficulty with RESRA link primitives:** As mentioned earlier, learners tend to create many summarative nodes on minor themes of a paper. Since RESRA link primitives are designed for representing the relationships between major themes, learners might have found it difficult applying these primitives to detailed nodes.
- **Difficulty in using CLARE's link mode:** The text-based interface of CLARE's link mode might have prevented some learners from using it.

Table 6.5: Distribution of summarization strategies

Strategies	Exp. A	Exp. B	Exp. C	Exp. D	Exp. E	Total
Sum	5	5	3	4	6	23
Sum \Rightarrow Link	6	1	1	3	0	11
Sum \Leftrightarrow Link	2	7	7	0	2	18
Sum \Rightarrow Link \Rightarrow (Sum \Leftrightarrow Link)	1	3	1	1	0	6
other	2	0	4	0	0	6

The majority of summarization was done through a single-round navigation of the source nodes, which might have in part accounted for the fact that, despite the large average number of summarative nodes created, major themes of a paper were still often missed. Discerning the major themes of a paper and the relationships between them often requires relating different parts of the paper, which is very difficult to achieve via a single pass of the artifact.

6.5.3 Evaluation

Table 6.6 shows four major strategies used by the learners when evaluating the content of research papers and the distribution of these strategies in various experiments. The data reveals that half of the learners did not do evaluation at all during exploration phase. Among the learners who did evaluation, the majority of them (84%) did so by mixing summarization and evaluation into a single stream of activity, i.e., Sum \Leftrightarrow evaluation or vice versa. There was no single instance in which a separate round was devoted solely to evaluation.

Table 6.6: Distribution of evaluation strategies

Strategies	Exp. A	Exp. B	Exp. C	Exp. D	Exp. E	Total
No evaluation	8	7	7	6	4	32
Sum \leftrightarrow Evaluation	5	8	3	0	1	17
Sum \Rightarrow Evaluation	1	0	0	1	2	4
Evaluation \leftrightarrow Sum	0	1	1	1	0	4
Missing	2	0	4	0	1	6

6.6 Collaborative learning using CLARE: a case analysis

This section presents a detailed analysis of a CLARE session involving a group of four learners⁵. The primary purpose is to illustrate the previous findings on the hypotheses, common representation issues, and usage strategies through a single CLARE session. The section begins with an overview of the research paper used in the current example, and a summary of the session to be analyzed. A prototypical RESRA representation of the research paper is provided to serve as a basis of comparison. The following two subsections present a detailed analysis of the results from the exploration and consolidation phases, respectively. The section concludes by comparing the result from the current CLARE session with the hypothetical scenario described in Chapter 1.

⁵The current data set consists of a total of 16 group sessions. The current group session was selected based on two primary criteria: a relatively even participation among the group members and a manageable number of nodes created during both exploration and consolidation phases. The true identities of the learners who participated in the current CLARE session are disguised to protect their privacy. For comparison purpose, the four pseudo-names from Chapter 1 are used instead. The complete CLARE session summary for the current case can be found in Appendix C.

6.6.1 Overview of the CLARE session

The artifact studied in the current example — “The Automated Requirement Traceability Systems (ARTS): an experience of eight years” by R.F. Flynn and M. Dorfman [FD90] — was from the designated readings for ICS613 (“advanced software engineering”). It represents one important type of research literature found in software engineering, i.e., *case studies*, which typically reports the experience related to technology or methodology adoption in a given organizational context. This specific paper describes an eight-year experience of developing and using a requirement management system called ARTS. The purpose is to describe ARTS as a state-of-the-art system of its type at that time. More importantly, it is to show that, to be successful in *inserting* a new system, such as ARTS, into a software development project, two factors must be taken into consideration: the flexibility of the tool itself and a good understanding of the application environment. A conceptual framework called “the tool-to-task paradigm” is proposed to help guide future efforts in this area. The main thematic components of this paper are captured through the following RESRA node instances:

- Poor practice in requirement management (**problem**): Despite its importance in developing large software systems, requirement management (RM), especially, traceability considerations, is still misunderstood and only rarely performed correctly.
- Critical success factors in RM systems (**claim**): To be successful in developing and using a RM system, two conditions must be met: (1) the RM tool must be flexible; and (2) the entire application environment must be taken into account.
- Reasons for the success of ARTS (**evidence**): Nine main factors accounting for the successful adoption of ARTS at Lockheed.
- Difficulties/problems with ARTS (**evidence**): Detailed list of complications experienced while developing and using ARTS.
- Tool-to-task-paradigm (**concept**): A way of understanding of the application environment of a software system that consists of four components: *situation, computer, people, and data*.

- Usefulness of the tool-to-task paradigm (**claim**): The tool-to-task paradigm provides a useful framework to guide the development of new RM tools and the application of existing RM tools to new contexts.
- The ARTS system (**thing**): The requirement management system successfully developed and used at Lockheed during the past eight years.

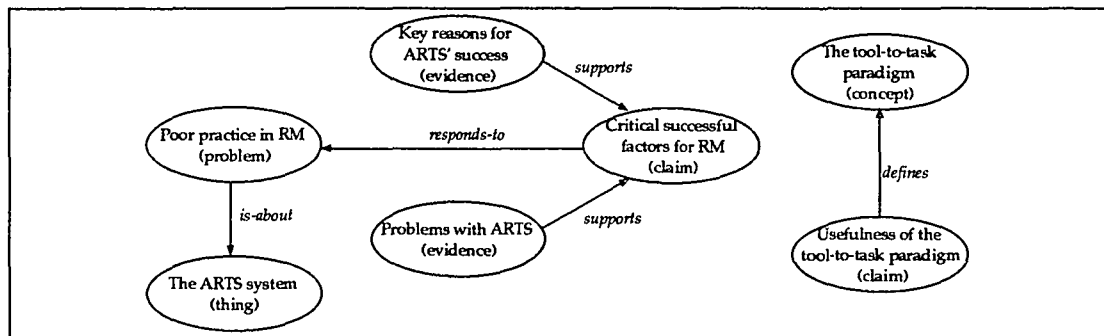


Figure 6.10: A RESRA representation of [FD91]

The relationships between the above RESRA nodes are depicted in Figure 6.10. As mentioned earlier, this paper is a *case study* whose stereotypical RESRA structure (i.e., CRF) is not defined in the current set of CRFs (see Section 3.4). Hence, no comparison can be made between the current representation and the standard CRF.

The example CLARE session was generated by a group of 4 first-time users from a graduate course in requirement engineering. The group created a total of 92 nodes over a period of 10 days: 58 of these nodes were from the exploration phase and the remaining 38 were from the consolidation phase. The distribution of the CLARE usage time by the group members is shown in Figure 6.11. The group is quite balanced in participation: each members devoted about 9 hours, two-thirds of which were spent in the exploration phase, and the remaining spent in the consolidation phase. The average usage time of this session (9hr) is almost twice as much as the overall average of the CLARE experiment (see Table 6.1) This discrepancy is perhaps caused by the fact that the current group were the first-

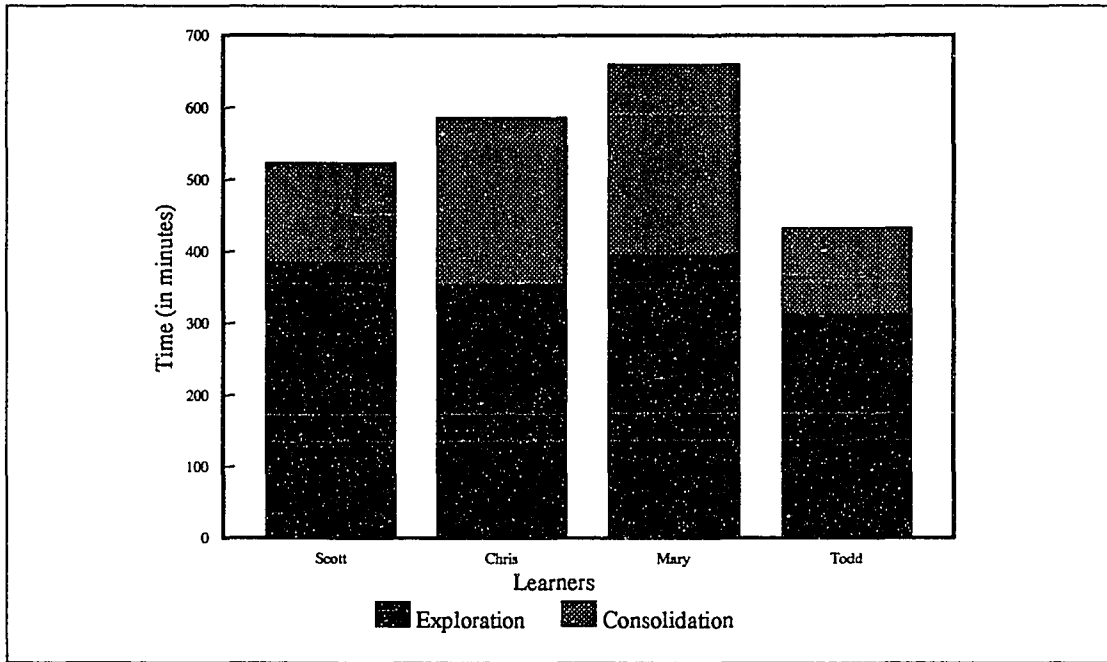


Figure 6.11: Distribution of the CLARE usage time

time CLARE users, who devoted part of their usage time to learning about the system. The following two sections present a detailed analysis of this session.

6.6.2 Analysis of the exploration result

Table 6.7 lists the distribution of the nodes created by the 4 group members during the exploration phase. Note that none of the group members created any evaluative node. Figure 6.12 shows a graphical representation of the 4 learner's nodes and links. Note that Todd did not have any links. Figure 6.13 shows the distribution of summarative nodes based on the source from which they are derived. Notice that Source Node 3 is the origin of 19 summarative nodes. Hence, it might be viewed as the hot section of the selected paper. In general, the four learners seem to have a great deal in common. As evident from the following sections, however, they also differ greatly in many other ways.

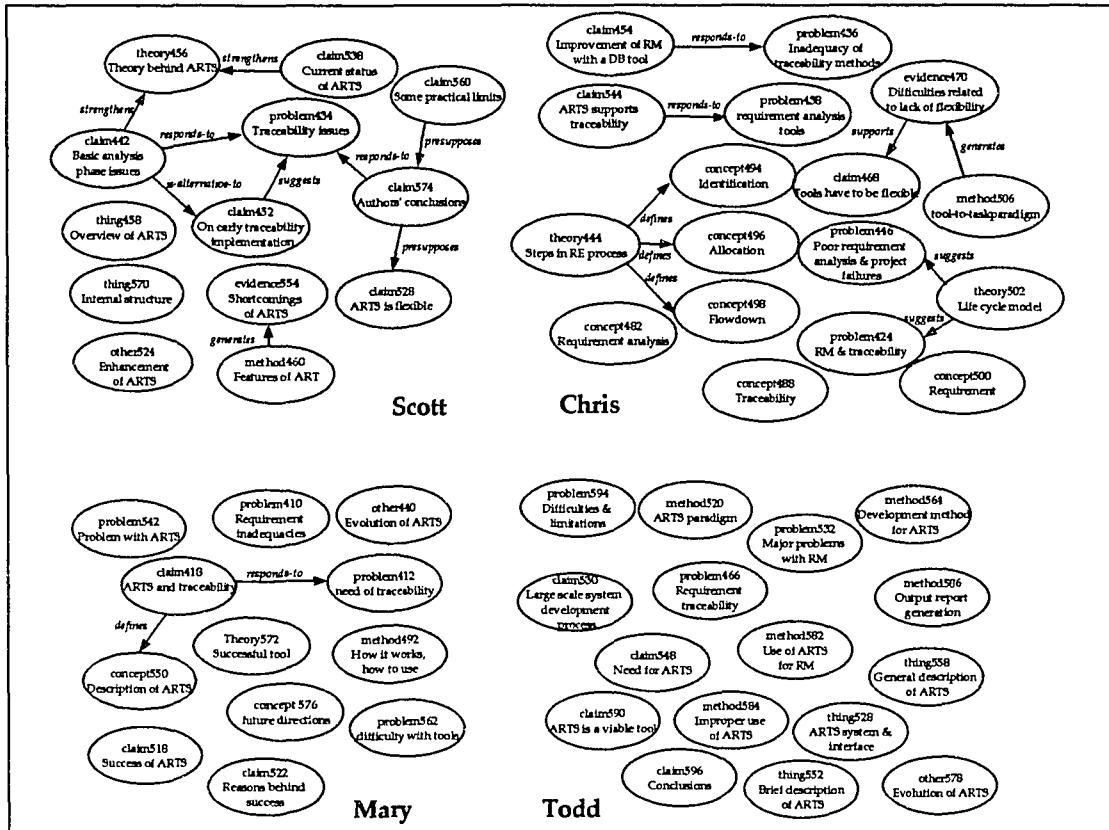


Figure 6.12: A comparative view of four RESRA representations of [Flynn90]

Table 6.7: Distribution of summarative nodes by Scott, Chris, Mary, and Todd

User	PR	CL	CO	ME	EV	TH	SO	TI	OT	Total
Scott	1	6	0	1	1	1	0	2	1	13
Chris	4	3	6	1	1	2	0	0	0	17
Mary	4	3	2	1	0	1	0	0	1	12
Todd	3	4	0	5	0	0	0	3	1	16

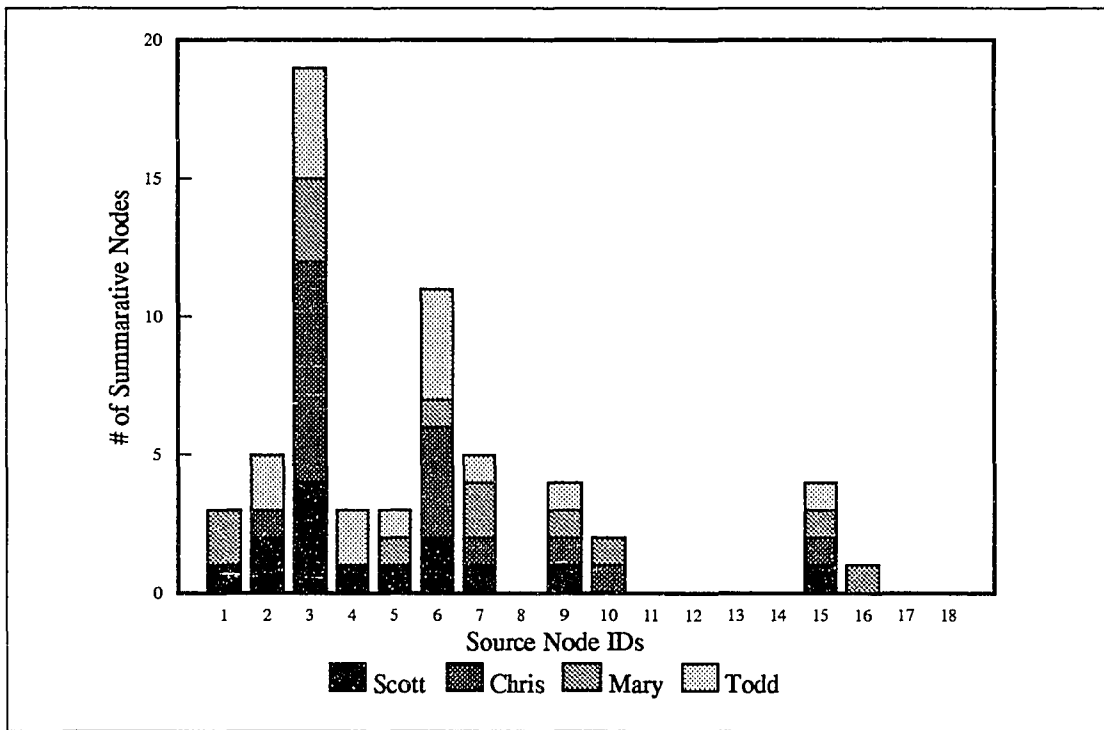


Figure 6.13: Distribution of summarative nodes over the source nodes

Scott

As shown in Table 6.7, Scott created 13 summarative nodes: 6 **claim**, 2 **thing**, 1 **problem**, 1 **theory**, 1 **evidence**, 1 **method**, and 1 **other**, respectively. The node “Traceability issues” (problem434), however, does not describe the main problem of the paper: the authors’ concern is not merely traceability issues but requirement management as a whole. Scott’s 6 **claim** nodes are “basic analysis phase issues” (claim442), “on early traceability implementation” (claim452), “ARTS is flexible” (claim528), “current status of ARTS” (claim538), “some practical limits” (claim560) and “author’s conclusions” (claim574). The current status of ARTS (claim538), as shown below, is not a **claim**. Furthermore, the lack of Traceability support in early RM systems (claim452) is in fact part of the **problem** the authors attempt to address, not a **claim** for which they provide supporting or countering evidence. The main thesis of the paper (see Section 6.6.1) is partially covered by “claim574,” i.e., “Author’s conclusions.” The claim that “ARTS is flexible” (claim528) presents a good example that shows the context-sensitive nature of RESRA primitives: at one level, it is a **claim** for which the authors provide supporting evidence, for example, ARTS allows its users to define their own record structures. At another level, “claim528,” along with its supporting evidence, is used as **evidence** to support a higher-level claim, i.e., the successful insertion of a RM tool into an existing project requires two conditions, one of which is that the RM tool must be flexible. Although the current paper fits to the latter, Scott’s representation took the former approach.

The node “Theory behind ARTS” (theory442), contains no “theory” about the ARTS system. Instead, it describes what the system was supposed to provide, e.g., requirement traceability. The other six nodes about different aspects of ARTS are: “overview” (thing458), “internal structure” (thing570), “features” (method460), “current status” (claim538), “enhancements” (other524), and “shortcomings” (evidence554). First, like

“theory442,” many of these nodes were incorrectly represented. For instance, while the authors *describe* the features and status of ARTS, the former was treated as `method` but latter, a `claim`. Second, quite an amount of overlapping is found between such nodes as “overview,” “features,” and “internal structure.” It is unclear why these nodes should be separate nodes instead of just one, as described in Section 6.6.1. One possible explanation might be that all these nodes were derived from different source nodes. Since Scott adopted an one-pass, linear approach to summarization, node creation is done on-demand without either forward or backward reference, which leads to overlapping and fragmentation.

Scott generated 9 RESRA tuples, which are depicted in the upper-left region of Figure 6.12. A number of these tuples are rendered invalid because the incorrect use of node primitives. For example, since the node “theory behind ARTS” (theory456) does not exist, the two *strengthens* links leading to this node are no longer valid. Similarly, the tuple “Features of ARTS (method460) $\xrightarrow{\text{generates}}$ “shortcomings of ARTS” (evidence554) is also spurious since “Features of ARTS” is not a `method`, as pointed out earlier. It is also unclear that “shortcomings of ARTS” is an `evidence` in the current context, since it is not connected to any `claim` node. The two links related to the node “On early traceability implementation (claim452), i.e., *is-alternative-to* and *suggests*, are also invalid because this node is not a `claim` but part of the `problem`. The remaining three tuples do somewhat capture the actual relationships that exist between these nodes:

- Author’s conclusion (claim574) $\xrightarrow{\text{generates}}$ traceability issues (problem434)
- Some practical limit (claim560) $\xrightarrow{\text{presupposes}}$ Author’s conclusion(claim574)
- Author’s conclusion (claim574) $\xrightarrow{\text{presupposes}}$ ARTS is flexible (claim528)

The above analysis of the summarization result by Scott shows a large number of in-

correct use of RESRA node and link primitives. This result demonstrates that (1) Scott did not have a good understanding of the semantics of RESRA, and/or (2) Scott did not have a good grasp of the main ideas of the paper. The latter is precisely one of the problems CLARE is intended to help Scott overcome. However, the achievement of the latter is contingent on a good grasp of the RESRA language. The other factor that might have accounted for the above problem is Scott's summarization strategy: the one-pass, linear approach does not seem sufficient to uncover the thematic structure of a paper; instead, it encourages Scott to create a representation that parallels the presentational structure of the paper.

Chris

Chris' summarization of [FD90] consists of 17 nodes: 4 `problem`, 3 `claim`, 6 `concept`, 2 `theory`, 1 `method`, and 1 `evidence`, respectively. Overall, these nodes together captured most of main thematic features of the paper. Compared to the prototypical and Scott's representations, both of which contain one general `problem`, Chris' four `problem` nodes ("RM and traceability" (problem424), "inadequacy of traceability methods" (problem436), "RA tools" (problem438) and "poor RA and project failure" (problem446)) are a finer differentiation of the problem that the original paper attempts to address. This detailed breakdown of the problem, however, requires `claim` to be handled in a similar fashion. Chris identified 3 `claim`: "ARTS supports traceability" (claim544), "Improvement of RM with a DB tool" (claim454), and "tools have to be flexible" (claim468). The first two are directly targeted at the first two problems identified earlier. Chris' two `theory` nodes, i.e., "steps in RE process" (theory444) and "life cycle model" (theory502), are not well-grounded, however. The first node, for example, describes the process steps of the requirement management at Lockheed. It should be a `method` instead of a `theory`, for the implementation

of that process would generate evidence to support or counter the assumptions underlying it. A similar comment can be made to the node “life cycle model” (theory502).

Chris also identified six **concept**: “requirement,” “requirement analysis,” “traceability,” “identification,” “allocation,” and “flowdown.” First, some of these concepts, noticeably, “requirement” and “requirement analysis,” are so commonly used in the current paper and group setting (i.e., graduate students in requirement engineering) that it is unclear why they should be singled out as something important. Second, concepts such as “identification” and “allocation” are an integral part of their “parent,” high-level conceptual construct (i.e., “Steps in the RE process” (theory44)). When treated separately, they begin to lose important part of their “identity,” and thus become less meaningful. Because of such dependency, it is preferred that these concepts be not treated as separate entities.

Chris generated only one **evidence** node: “lack of flexibility in tool (ARTS) leads to a number of difficulties in their use” (evidence470). The node is significant because, in the previous section, Scott identified a **claim** named “ARTS is flexible.” It does not seem very likely that the paper makes a claim that ARTS is flexible while at the same time, shows the difficulties in using the system because of its inflexibility. The contradiction must come from the interpretation. In fact, the picture about what the authors really attempt to convey becomes clear when looking closely at the tuple relationship of the current evidence: “ARTS’ lack of flexibility and resulted difficulties” (evidence470) $\xrightarrow{\text{supports}}$ “Tools have to be flexible” (claim468). In other words, the author uses the flexibility and inflexibility of ARTS and resulted benefits/difficulties as **evidence** to *support* the claim that requirement management tools have to be flexible. It is apparent that Scott mis-constructed the intended thesis of the paper.

Mary

Mary's summarization of [FD90] consists of 12 nodes: 4 **problem**, 3 **claim**, 1 **concept**, 1 **method**, 1 **theory**, and 1 **other**, respectively. Of the 4 **problem**, the node "Need of traceability" (problem412) is similar to Scott's "traceability issues" (problem434) and Chris' "RM and traceability" (problem424) and "Inadequacy of traceability methods" (problem436). The node "requirement inadequacies" (problem410) describes the bottleneck of requirement specification and management in light of the changing needs of the user. It is similar to Chris' "poor requirement analysis and project failures" (problem446), and parts of Scott's "basic analysis phase issues" (claim442). The other two problems are "problem with ARTS" (problem542) and "difficulty with tools" (problem562), both of which are in fact described by the authors. However, it is unclear that they are intended "focal points" of the paper.

Like Scott, Mary had several nodes that describe different aspects of ARTS: "ARTS and traceability" (claim418), "evolution of ARTS" (other440), "success of ARTS" (claim518), "reasons behind success" (claim522), "problem with ARTS" (problem542), "how it works, how to use" (method492), and "description of ARTS" (concept550). Most of these nodes used incorrect RESRA node types. For instance, the node "ARTS and traceability" (claim418) describes how traceability is supported in ARTS. It is similar to "how it works, how to use" (method492) and "description of ARTS" (concept550). They can be merged into a single node. The resulted node, based on the RESRA definitions, will not be a **claim**, nor a **method**, but a **thing**, for ARTS is a physical object under study. The nodes "success of ARTS" (claim518), "reasons behind success" (claim522), and "problem with ARTS" (problem542) are evidence for supporting the claim that the successful development and use of requirement management tools require that (1) the tool is flexible, and (2) the entire application environment is taken into account. The node "future directions"

(concept576), in fact, contains a number of less substantiated claims that are intended for future researchers on the subject. An example of such claims is that hypertext is a promising technology for enhancing requirement traceability.

Todd

Todd's representation of [FD90] consists of 16 nodes: 3 `problem`, 4 `claim`, 5 `method`, 3 `thing`, and 1 `other`. The three `problem` he identified are: "requirement traceability" (problem466), "major problems with RM" (problem532), and "difficulties and limitations" (problem594). The first two nodes are roughly equivalent to Chris' "RM and traceability" (problem424) and the combination of "inadequacy of traceability methods" (problem436) and "requirement analysis tools" (problem438), respectively. As described earlier, the theme "difficulties and limitations" (problem594) is used by the authors as counter-evidence to support their primary claim. Hence, in the current context it should be an `evidence` instead of a `problem`. Two of the four `claim` nodes created by Todd are considered incorrect. The node "large system development process" (claim530) presumably refers to the life cycle model which, as pointed out earlier, is a `method`. The "need for developing ARTS" (claim548) is only briefly mentioned as part of the introduction to the system, and is therefore too peripheral to be considered as a `claim`. Most of the five `method` nodes are also questionable in conforming to the semantics of the primitive. For example, the node "output report generation" (method586) discusses the flexibility of report generation in ARTS. Hence, it should be represented as `evidence` to support the main claim. Similarly, the node "improper use of ARTs" (method584) indicates that a tool is often used in unintended ways. As a result, the design of the tool should take into account the entire application environment. The latter problem is precisely one of the two components of the main thesis. Hence, the current node may also be viewed as `evidence` instead.

The three **thing** nodes: “brief description of ARTS” (thing552), “general description of ARTS” (thing558), and “ARTS system and interface” (thing528), describe different aspects of the same physical object, i.e., ARTS. They may be merged into a single **thing** node called “the ARTS system,” just as in the prototypical representation. Since Todd created no tuples in his summarization, his perspective on how the above nodes are related to one another cannot be assessed.

Summary

The above analysis of the summarization result of [FD90] by Scott, Chris, Mary, and Todd demonstrates a wide variety in the learners’ understanding of both the content of the paper and the representation language. It also shows major differences and similarities between the four representations and points of view. By comparing these representations with the example described at the beginning of this section, the following observations can be made:

- All four learners were all very close in identifying the central problem of that paper. However, none of them was able to clearly articulate the authors’ primary claim.
- The representations from the group as a whole are finer-grained than the prototypical one, although individual representations differ greatly in their focus (e.g., Chris’ **problem**, Todd’s **evidence**).
- The representations from the group as a whole show little synthesis of thematic elements from various parts of the paper. Instead, the representations bear a close relationship with the linear, presentational structure of the paper.
- The learners do not seem to have a good grasp of the semantics of RESRA, as shown by the large number of incorrect use of both node and link primitives.

The following section presents the analysis result from the consolidation phase of the current example.

6.6.3 Analysis of the consolidation result

Table 6.8: Distribution of argumentative nodes by Scott, Chris, Mary, and Todd

User	Claim	Evidence	Critique	Question	Suggestion	Total
Scott	0	0	0	3	3	6
Chris	0	1	3	1	3	8
Mary	1	0	10	2	4	17
Todd	0	0	1	1	1	3

Table 6.8 shows the top-level summary of the argumentative nodes created by Scott, Chris, Mary, and Todd. Of a total of 34 nodes created during this phase, 32 of them are *evaluative* nodes, which indicates that argumentation is dominated by collaborative evaluation of the representations generated from the previous phase. Figure 6.14 shows a detailed picture of all summarative nodes and links from Phase I and the argumentative nodes directed to them.

Mary is the most active member in the group; she created 17 nodes during the current phase. Todd is least active; he created only 3 nodes. In terms of the attention received, Scott received the most attention: 11 argumentative nodes (31%) were targetted at his representation. On the other hand, Mary received the least attention: only 6 argumentative nodes (17%) came to her way. It is interesting to note that Scott's summarative representation, as described in the previous section, is among the poorest in the group in terms of distortions, incorrect use of the primitives, and spurious links but he received the most attention in Phase II. The seemingly existence of an inverse relationship between the quality of summarization and the level of contribution to *triggering* group discussion in Phase II

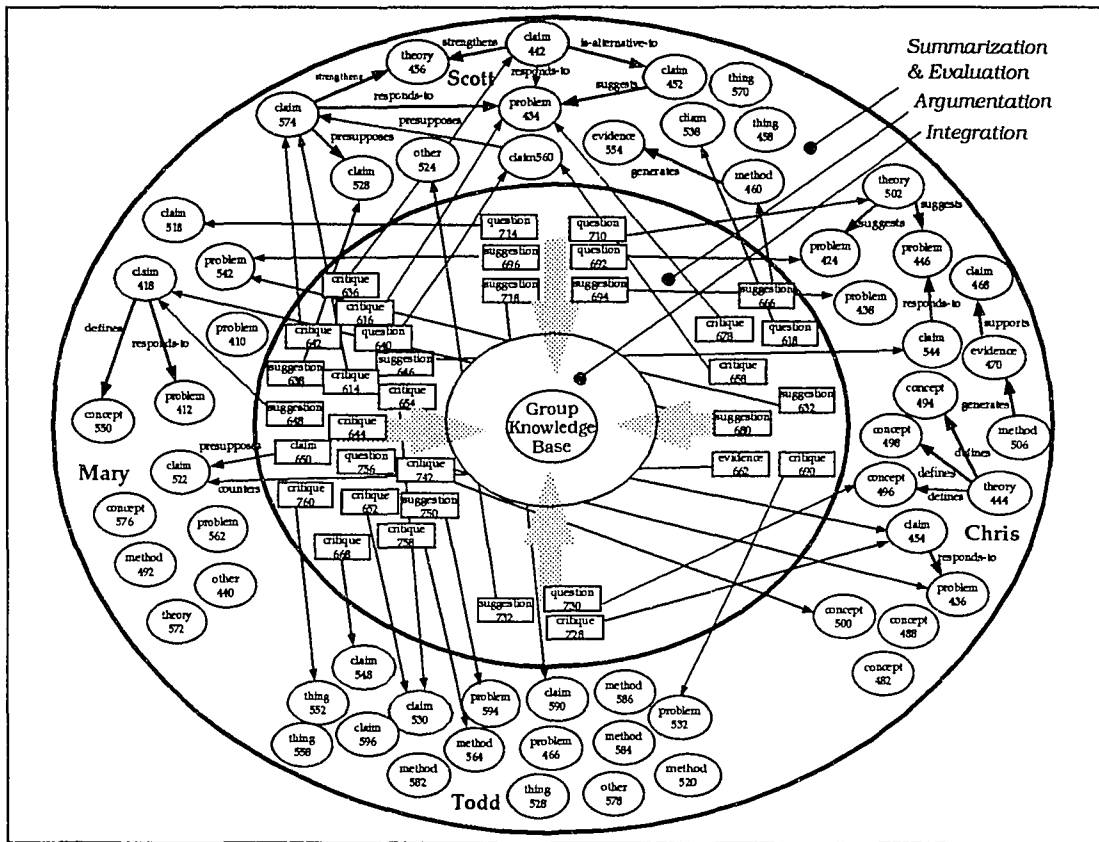


Figure 6.14: Overview of summarization and argumentation by Scott, Chris, Mary, and Todd

has several interesting implications. However, from this single instance it is difficult to conclude that such a relationship indeed exists.

Analysis of argumentative nodes

Scott. Scott created 6 nodes during argumentation: 3 **questions** and 3 **suggestion**, respectively. Of the 3 **questions** nodes, 2 of which are in fact **question** nodes: “Re: life cycle model” (question710) and “Re: success of ARTS” (question714). In the former Scott suggested to replace the “Waterfall” model by some newer models (e.g., “incremental model”) to ease certain requirement management problems. In the node “Re: success of ARTS,” Scott proposed to use standard benchmarks to quantitatively evaluate various RM tools.

Chris. Chris’ contribution at Phase II consists of 8 nodes: 3 **critique**, 3 **suggestion**, 1 **question**, and 1 **evidence**. The last node deserves special attention since it belongs to a separate category of argumentation called *constructive* argumentation, as opposed to *evaluative* argumentation to which most of the current phase belongs. In the constructive argumentation, the learner does not merely critique or question another learner’s position, he in fact engages in active knowledge-building by formulating new problems, proposing alternative claims, supplying new evidence, and so on. In the current example, Chris created an **evidence** node named “ARTS is not at all that flexible” (evidence662), which contains the reference to another **evidence** node he created in Phase I (i.e., “difficulties related to lack of flexibility” (evidence470)), to counter Mary’s **claim** that ARTS is successful because it is flexible (“Reasons for success (claim522)).

In other argumentative nodes, Chris proposed various ways to improve ARTS, includ-

ing the use of 4GLs and distributed database managers. In “Can ARTS be transferred to other platforms?” (question618), he pointed out the ambiguity of a sentence used by Scott in his node “current status of ARTS” (claim538). Of a total of Chris’ 8 argumentative nodes, 4 of which are targeted at Scott, 1 at Todd, and the remaining 3 at Mary.

Mary. Mary’s contribution to the group during the consolidation phase consists of a total of 16 nodes: 10 **critique**, 2 **question**, 4 **suggestion**, and 1 **claim**. Her critiques can be grouped into two main categories: pointing out the incorrect use of RESRA primitives and identifying ambiguities/inaccuracies in other learners’ representations. The former includes “requirement problem” (critique636), and “tree format” (critique644), “methodology” (critique654), “requirement?” (critique558), and “evolution” (critique758). In the last node, for example, Mary pointed out that Todd’s treatment of “evaluation of ARTS” as a **method** is incorrect by quoting the RESRA’s definition of **method**. Similarly, in “methodology” (critique654), she pointed out that Todd’s “large software development process” (claim530) should be a **method** instead of a **claim**. In “requirement problem” (critique636), Mary argued that Scott was wrong: the inadequacy of the analysis phase should be a **problem** instead of a **claim**.

The second group of **critique** nodes include “conventional traceability method” (critique742), “ARTS’ adaptability” (critique642), “incomplete description” (critique614), and “over-statement” (critique616). In the last node, for example, Mary pointed out that the author did not say that the success of traceability systems is “proportional” to their adaptability, as stated by Scott in his “traceability issues” (problem434). In the node “ARTS’ adaptability” (critique642), Mary also indicated that Scott was wrong in stating that ARTS is adaptable to various application environment. She also pointed out that the fact is just opposite — ARTS mainly runs on DEC VAX and is not flexible. The process data shows

that, to assess the accuracy of Scott's representation and create critique nodes such as the ones above, Mary in fact verified the representation content against the source from which Scott's problematic nodes were derived.

Todd. Todd's contribution in Phase II consists of 3 nodes: "requirement management and tree structure" (critique728), "hierarchical allocation" (question730), and "development platform" (suggestion730). The first two nodes were directed to Chris. The third was directed to Scott.

6.6.4 Discussion

The above analysis of an actual CLARE session forms a contrast with the hypothetical scenario described in Chapter 1. Although the two share the same objectives (i.e., collaborative learning from research papers) and the same environment (i.e., CLARE), the outcome was quite different, as shown in Figures 6.14 and 1.5. At a top-level, some noticeable differences between the real and hypothetical CLARE sessions are:

- The outer layer of Figure 6.14 contains a much larger number of summarative nodes;
- The outer layer of Figure 6.14 contains no evaluative nodes;
- Figure 6.14 does not show the *comparison* layer; and
- The *integration* layer in Figure 6.14 does not contain any nodes.

At a deeper level, the following differences were found between the two CLARE sessions:

- The two Scotts used a different summarization strategy. In Chapter 1, the hypothetical Scott adopted a "two-round, Sum \Leftrightarrow Link" strategy. The real Scott used a

“single-round, Sum \Rightarrow Link” strategy. Moreover, the hypothetical Scott invoked the Template Guide function. He also used CLARE’s advice to guide his creation of RESRA tuples during the late stage of summarization. The the real Scott did not use either functions.

- In the hypothetical session from Chapter 1, all group members seemed to have little problem with the semantics of RESRA. As shown above, in the real session each student had different interpretations of RESRA.

The above differences between the hypothetical scenario and the actual CLARE session illustrate some main usage gaps between them:

- In a real CLARE setting, most important themes of a paper were not always represented. Quite often, many minor themes were included;
- CLARE’s instrumentation does not allow one to determine which two nodes a user was comparing;
- Evaluation is not always performed during the exploration phase;
- Integration is not always performed in the consolidation phase;
- CRF function were under-utilized by many users; and
- Learners had different connotative interpretations of the same RESRA primitives.

In addition to these discrepancies, several other observations can also be made. First, most of the learner’s attention was still devoted to the bootstrapping stage of the collaborative learning process, i.e., summarization and evaluation; 66% of the group member’s time was spent in the exploration phase, which does not even include evaluation. Second, the example also indicated that, even as a group, the main thematic structure of a paper is

always reconstructed. And finally, the collaborative potential of the RESRA language was limited by the lack of shared understanding of its semantics and the resulting inconsistent usages.

6.7 Summary and conclusions

Summary

This chapter discusses main findings from a set of experiments on CLARE. Results from these experiments have in general confirmed (1) the viability of CLARE as a novel environment to support collaborative learning from scientific text, and (2) the usefulness of RESRA as a representational basis for such an approach. They have also revealed a number of problems about the system:

- At the representational level, RESRA is interpreted in many different ways. It is also often used incorrectly. Nevertheless, the majority of learners still view RESRA as a useful and novel tool. This apparent inconsistency between the two (called “the RESRA paradox”) is discussed below.
- At the implementational level, the user interface, in particular, the link mode, is the greatest barrier to effective use of CLARE; and
- At the experimental level, inadequate training on RESRA and CLARE, along with other factors such as tight schedules and improper choice of research papers, seems responsible for most of user dissatisfaction.

The remainder of this section discusses several specific issues that arose from the experiments.

The “RESRA paradox”

As reported in Section 6.3, 84% of learners indicated that RESRA provides a useful means of characterizing the content of scientific text, and 90% of learners agreed that RESRA helps expose different points of view on an artifact. On the other hand, an analysis of the CLARE database reveals a great deal of incorrect use of RESRA (see Sections 6.4 and 6.6). The latter might be attributable to a number of factors, including insufficient training and online examples, and a lack of detailed usage guidelines.

The apparent gap between positive user attitudes about RESRA and incorrect use of the representation has several important implications. First, the utility of RESRA is not necessarily the same as the *correctness* of its usage. In other words, a learner may use a primitive incorrectly but still find it useful, since it helps identify what would have been missed otherwise. Such things can happen because of the heuristic value of RESRA, which exists independent from the correctness in the interpretation of the primitive. Second, because of the abstract or meta nature of RESRA, and the differences in backgrounds, perspectives, and interests among different learners, mis-interpretations and incorrect usages might be inevitable, regardless the amount of training received. In fact, representation-level errors might also help expose deep-level gaps in the learner’s knowledge structure, and thus create opportunity for potential collaboration among learners. However, a large number of representation-level errors also limit the potential of RESRA as a collaborative learning framework. Section 8.3 identifies several extensions to CLARE and RESRA that will help reduce the number of RESRA usage errors.

Mapping from text to representation

The experimental results also revealed that the mapping from scientific text to a representation such as RESRA is not always straight-forward. As illustrated in Section 6.6, different

learners have quite different and, in many cases, mutually incompatible representations of the same artifact. A number of factors might have accounted for these variations:

- *Lack of consistent interpretation of RESRA (see above):* This language-level inconsistency leads to inconsistent use of RESRA, which in turn leads to inconsistent representations.
- *Absence of RESRA usage guidelines:* The lack of explicit, detailed guidelines on issues such as the granularity of representation, *major themes* versus *minor themes*, and *learner's views* versus *author's views* might be responsible for a sizable portion of the representational variations.
- *Different interpretations of the artifact itself:* Because of the difference in their backgrounds, perspectives, interests, and so forth, different learners have different interpretations of the same artifact, regardless the type of representation scheme being used.

Because of the dynamic interplay among the above factors, it does not seem surprising that wide variations are found between different learners' representations. However, the effectiveness of CLARE as a collaborative learning environment is contingent upon a basic, shared understanding of RESRA among learners, and a certain level of comparability between different learner's representations of the same artifact. Such a level of consensus has not yet achieved in the current evaluation experiment. Section 8.3 describes specific proposals for alleviating this problem.

Relationships with other theories and systems

The CLARE findings described earlier (see Section 6.3) confirm that meta-cognitive tools such as RESRA facilitate *meaningful learning* (see the user comments in Section). They

also show that the constructionist view of learning as collaborative knowledge-building is a sound theoretical framework for building collaborative learning systems. The next chapter reviews these two conceptual formulations: constructionism and cognitive learning theory. In addition, it also compares CLARE with other collaborative learning systems, such as virtual classrooms and hypermedia systems. Chapter 8 elaborates a number of directions in which CLARE can be enhanced.

Chapter 7

Related Work

CLARE represents a confluence of several streams of research spanning across a number of intellectual disciplines: computer-supported cooperative work (CSCW), human learning, knowledge representation, sociology of knowledge, and hypertext. While this dissertation falls under the broad umbrella of CSCW, the theoretical motivation and practical application reside in human learning.

This chapter is organized into four sections. Section 7.1 reviews the theoretical work on which CLARE is based. More specifically, it covers constructionism and the assimilation theory of cognitive learning. Section 7.2 describes schema theory and related knowledge representation schemes. Section 7.3 surveys major existing collaborative learning systems and related empirical findings. This chapter concludes with a summary of the relationships between CLARE and the work being reviewed.

7.1 Theoretical underpinnings

CLARE is grounded in two main theoretical tenets: constructionism and assimilation theory of cognitive learning. The former provides CLARE a philosophical foundation. The latter serves as an overarching pedagogical framework. The following sections provide an overview of both.

7.1.1 Constructionism

Constructivism, succinctly put, views that knowledge is *constructed* rather than merely *acquired* by the learner. It is a mode of learning in which the student plays an active, contributing part rather than being a passive target of knowledge transmission. To a constructivist, the learning environment is more of a give-and-take, with the teacher being just one of many resources upon which the student can call. Constructivism is often associated with the theories of Jean Piaget, who contends that restructuring of prior knowledge (learning) requires challenging existing views and coordinating old with new knowledge [Pia77], and that these conditions is present when learners interact with peers of differing but also inadequate views [Pia32].

Constructionism, which represents one variation of constructivism, holds that knowledge, especially scientific knowledge, is socially constructed [BL66, KC81]. Such knowledge is not the same for individuals but is *taken-to-be-shared* [RR92] with communities of learners. To become a member of such a community, students need to actively engage in interactions and undergo learning situations which allow them to immerse into the discourse practice of a field. In order to to form classroom communities which function like those of scientists, for example, students need to have opportunity engaging in authentic practice of scientists.

CLARE embodies the constructionist view in two ways. First, it treats learning not merely as reading or understanding what a research artifact says but as knowledge construction which benefits from explicit process- and representation-level support. Second, CLARE is built on the premise that knowledge-building among learners bears much resemblance with knowledge-building in the scientific community. Learners can gain understanding of how scientists construct knowledge by systematically studying the artifacts they generate. Subsequently, they may apply that knowledge to improve their own

knowledge-building practice. RESRA, SECAI, and CLARE were designed to guide and facilitate this process.

7.1.2 Assimilation theory of cognitive learning

The *theory of meaningful learning*, also known as the *assimilation theory of cognitive learning*, is an important theoretical framework in educational psychology. It has evolved and been tested at Cornell University over the past three decades [Aus63, NG84]. The thrust of this theory is the emphasis on meta-learning, that is, *learning how to learn*, and the role of the meta-knowledge in human learning. The basic tenets of this theory include:

- The single most important factor influencing human learning is what the learner already knows, i.e., prior knowledge;
- Learning is evidenced by a change in the meaning of experience rather than a change in behavior — a view that is long held by behavioral psychologists; and
- The key role of the educator is to help students reflect on their experience (and hence give it new meanings), and construct meanings from the artifact in light of the changing experience.

The theory introduces two important concepts: *progressive differentiation* and *integrative consolidation*. The former states that, since meaningful learning is a continuous process wherein new concepts gain greater meaning as new relationships are acquired, concepts are never finally learned, but their meanings are constantly revised, and made more explicit as they become progressively more differentiated. The latter refers to the fact that meaningful learning is enhanced when the learner recognizes new relationships between related concepts and propositions. To assess what a learner already knows and how it changes over time, Novak and Gowin [NG84] propose two meta-cognitive tools,

called *concept maps* and *Vee diagrams*, both of which are described in the following section.

The above view that human learning is *meaning-making* places knowledge representation at the focal point of the human learning process, since knowledge representation defines not only the form in which a certain type of knowledge is highlighted to the learner, but also the process by which such a form is derived. Moreover, knowledge representation languages, which are called *meta-cognitive tools* by educational researchers, are the standard language for characterizing both knowledge structures and corresponding cognitive structures. They help the learner differentiate and organize newly acquired meanings. The next section describes major related work in this area.

7.2 Representation and human learning

This section briefly describes what *schema theory* is and how it is related to CLARE. In addition, it also compares RESRA with a number of alternative representation languages, namely, IBIS, Toulmin's model of argumentation, concept maps, and Vee diagrams.

7.2.1 Schema theory, representation, and learning

One important conceptual framework in cognitive psychology is *schema theory*, which posits that human minds store and retrieve knowledge about the external world in terms of abstract categories called *schemas* [SFG⁺87]. A *schema* is defined as “a data structure for representing generic concepts in memory” [Rum80]. As such, a schema forms “a building block of cognition” that affects how new information is absorbed as well as how old information is retrieved from memory. Alba and Hasher [AH83] identify four main functions of schemas:

- *Selection*: Filtering out certain types of information before passing on to memory representation.
- *Abstraction*: Information reduction by omitting certain types of details.
- *Interpretation*: Inferring missing information based on previous instances of the schema.
- *Integration*: Grouping together related or similar information.

Although schema theory is a psychological framework for explaining how humans construct meanings from written or spoken words, it has been heavily influenced by work in artificial intelligence [Abe81, SA77]. In fact, *script* — the schema about activities and processes — is used both as a knowledge representation scheme for AI programs and in psychological research.

Schema-based approaches have been used to study expert-novice differences in reading and understanding text. It is found that novice readers tend to skim the text and retain isolated facts. Skilled readers, on the other hand, recognize patterns/schemas that relate different parts of the text into a coherent whole. Their strategies include searching the text for its underlying structure, identifying the major text schema, and formulating relational links between major and subordinate ideas [vDK83, VTY83].

Schema theory and the above empirical findings are directly relevant to CLARE: RESRA, for example, might be viewed as a set of schemas for characterizing the deep structure of scientific text. In particular, RESRA tuples and CRFs serve similar purposes as other types of schemas, such as *scripts*, in helping learners understand the content of scientific text through *selection*, *abstraction*, *interpretation*, and *integration*. The expert-novice differences in their use of schemas invite similar studies to be conducted in the CLARE environment (see Section 8.3).

7.2.2 RESRA and other representation schemes

Although RESRA is unique as a conceptual framework for characterizing thematic features of scientific text and for facilitating collaborative learning, the use of semi-structured representation in ill-defined tasks is not new. A number of such schemes have been invented and used in domains such as software design [LL91]. This section reviews two of such representations: *IBIS* and *Toulmin's model*. In addition, it also discusses two similar approaches that are proposed to specifically support human learning: *concept maps* and *Vee diagrams*.

IBIS. IBIS, which stands for *Issue-based information systems*, was originally proposed by [KR77] for deliberating design decisions in information systems. There are several variations of this representation, one of which, called *gIBIS* ([CB88]), is shown in Figure 7.1.

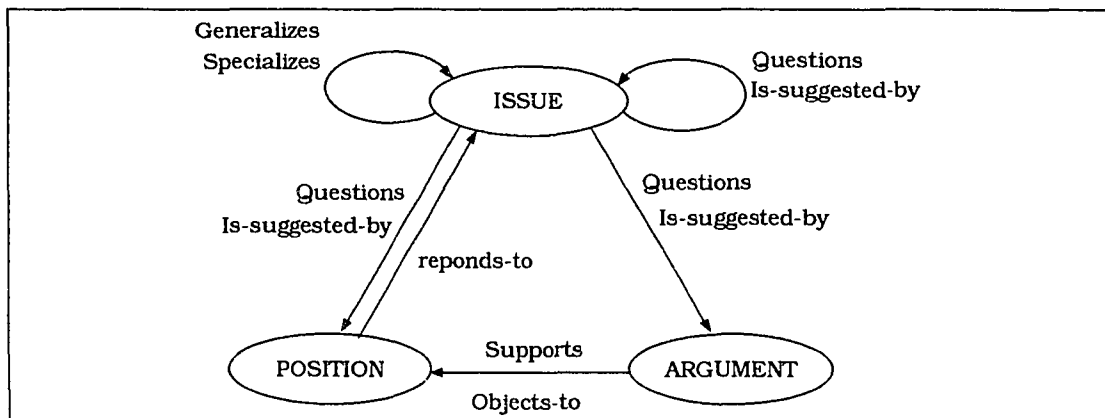


Figure 7.1: IBIS model of argumentation (based on [Conklin87])

The main feature of IBIS is that it is parsimonious: the three node and seven link types can be easily learned. However, the limitation of this representation is also evident: the small set of primitives are not sufficiently expressive for many task domains. Furthermore,

the model is biased toward controversies. For example, it omits questions that are not deliberated in favor of those questions with which debate and controversy are likely to be associated [MYBM91]. RESRA was originally built on the gIBIS representation. In fact, features shown in Figure 7.1 are also found in RESRA. For example, *issue*, *position*, and *argument* are subsumed by `problem`, `claim`, and `evidence` in RESRA, respectively.

Toulmin’s rhetorical model. Figure 7.2 shows another widely used argumentation model proposed by philosopher Stephen Toulmin [Tou58]. The original purpose of this model was for delineating logical structure of an argument, although it is also useful for analyzing scientific controversies (e.g., [CSGMW92]).

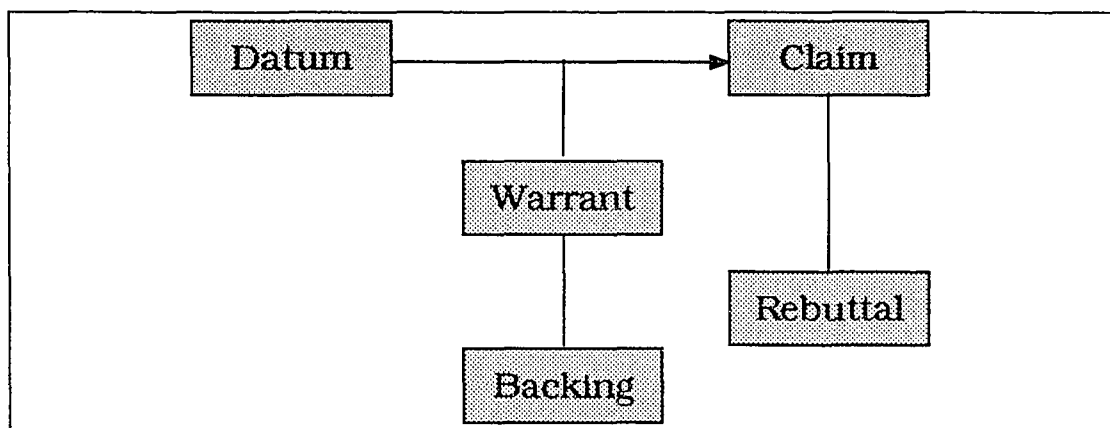


Figure 7.2: Toulmin’s model of argumentation (based on [Toulmin52])

Toulmin’s model suffers from similar problems as IBIS when placed in the CLARE application domain, that is, it is overly coarse-grained. The structure of scientific text varies widely, as shown the example CRFs. Using a rhetorically-based representation to characterize such structure is not often possible. Elements of Toulmin’s model, however, can also be found in RESRA, for example, `evidence` (*Datum*, *Backing*), and `claim` (*Claim*, *Rebuttal*).

Concept maps and Vee diagrams. Concept maps and Vee diagrams are two metacognitive tools proposed by educational theorists to (1) assess what the learner already knows; (2) discern changes over time in the learner's knowledge structure; and (3) facilitate *meaningful learning* [NG84]. The effectiveness of concept maps in accomplishing these goals seems well supported from field studies [Cli90, Nov90, RR92, ADS84]. Nevertheless, as a representation scheme for supporting collaborative learning, concept maps are not adequate (see Section 2.1.4). RESRA is a direct response to such inadequacies.

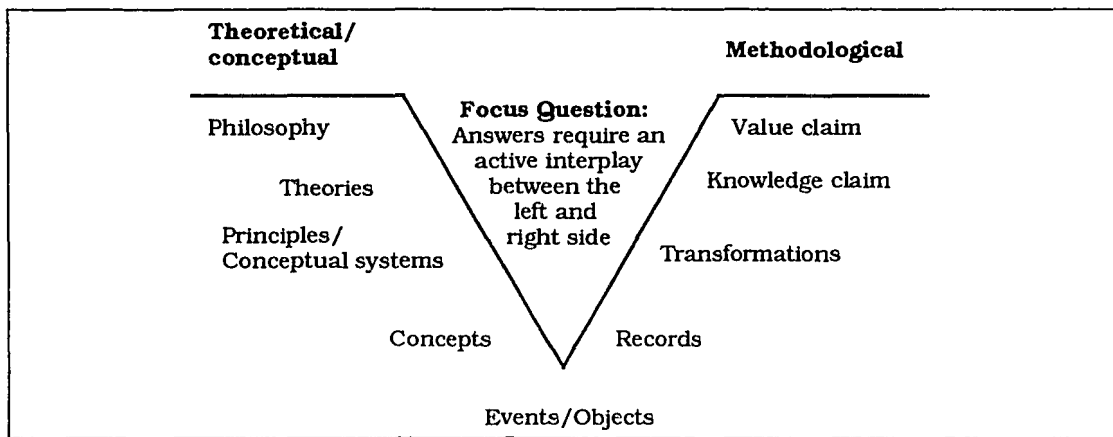


Figure 7.3: Novak and Gowin's Vee Diagram (from [NG84])

Figure 7.3 depicts a simplified version of the Vee diagram for understanding the nature of knowledge and knowledge production. The left side of the Vee is the *thinking-side*, while the right side is called *doing-side*. The two sides are linked together by the event/object that is under study. To answer the focus question in the middle requires an integration of issues raised from both sides. At a conceptually level, the Vee diagram offers an elegant and powerful means of exposing deep-level structure of knowledge: by constructing a Vee diagram for each problem situation, a learner can see inter-relationships among different knowledge components and gaps between them, if any.

The Vee diagram does not meet all important requirements of CLARE, however. First,

Vee was originally designed to help students and teachers better understand the nature of science laboratory work. In that setting, it is possible to adopt a *bottom-up* approach to knowledge, starting from the actual event/object under study. This approach, however, is not applicable to CLARE, since CLARE treats scientific text as the primary source of knowledge. Because scientific artifacts are *episodic* in nature [Swa90], artifact **A** may cover only the left side of the Vee, or even only the upper left portion of the Vee, while artifact **B** may cover the right side of the Vee. As a result, it is not always possible, nor necessary, to construct the entire Vee, in order to understand the content of the selected artifact.

Second, the Vee diagram does not lend itself to computerized support. Unlike RESRA, whose structure is consistent with a hypertext data model and is easily amenable to automated support, the Vee's symmetric, graphical approach makes it difficult to leverage through computerization.

Despite the above incompatibilities, the Vee diagram provides some useful heuristics for the refinement of RESRA. For example, the distinction between *knowledge claim* and *value claim* seems applicable to RESRA as well. Moreover, RESRA currently does not support the increasing level of *abstractness*, as one moves from the bottom to the top of Vee.

7.3 Collaborative learning systems: technologies and outcomes

This section surveys a number of important collaborative learning systems. For each system, it describes its major features and related empirical evaluation, if any. The section is organized into four parts, corresponding to four major types of collaborative learning sys-

tems: *virtual classroom, collaborative writing, hypermedia, and collaborative knowledge-building.*

7.3.1 Virtual classroom systems

The term *virtual classrooms*, or VCs, is used in a broad sense to encompass both general-purpose computer-mediated communication (CMC), such as e-mail, electronic bulletin-board systems, and specialized communication-based learning systems, such as EIES. The latter also provides such functions as instruction management tools (assignment tracking, grading, etc.) [Hil88]. VC systems possess the following main features:

- *Access orientation.* VC systems enable learners to transcend the geographical and temporal limitations of face-to-face meetings and allow them interact with one another asynchronously.
- *Affordable technology.* VC systems do not require sophisticated hardware and software technology. An inexpensive home computer equipped with a modem and communication software is often sufficient.

The combination of the above two features has made VC one of the most successfully and pervasively used collaborative learning systems. The remaining section summarizes empirical findings on one virtual classroom system called EIES [Hil88].

Findings on EIES. EIES (Electronic Information Exchange System), developed at New Jersey Institute of Technology, represents one of the earliest large-scale studies of the impact of computer conferencing and computer-mediated communication on student learning. The study involves a series of college-level courses, with subject matters ranging from introductory sociology, statistics, to computer science. The objective is to describe

the learning experience and outcomes of the VC delivery mode in relation to the traditional classroom and to determine conditions associated with good and poor outcomes. Some major findings from this study are:

- No consistent differences were found in scores measuring mastery of material taught in the virtual and traditional classrooms.
- Students in virtual classrooms showed more active participation.
- For those who participated regularly in VC, the level of interest tended to be high.
- Virtual classrooms provide more convenient access to educational experiences.
- Students also found virtual classrooms more time-consuming and more demanding, for they were required to play a more active part in the class.

While the quantitative results in most cases are inconclusive, the qualitative outcomes from this study show that, among well-motivated students, virtual classrooms provide a new opportunity to participate in different kinds of learning experience that is based on a community of learners working together to explore the subject content of a course.

Virtual classrooms and CLARE. Unlike virtual classroom systems which in many cases are often used *in place of* traditional classrooms, CLARE is designed to *complement* the face-to-face mode of learning. First, CLARE supports only one specific type of learning activity — collaborative study of scientific text. Other activities, such as lectures, are still conducted in the traditional mode. Second, the level of services provided by CLARE goes beyond access by overcoming the *representational* constraint of traditional mode of paper studying. CLARE makes the use of such a representation not only viable but also measurable. The latter can lead to a continuous improvement of both the process

and the representation. It is interesting to note that, even given the relatively simple functionality of VC systems, learners still find it time-consuming and demanding. Thus, it is not surprising that CLARE users experience similar problems (see Section 6.3.3).

7.3.2 Collaborative writing systems

Writing is an integral part of learning that often requires collaboration among different learners. The use of computers to support collaborative writing is quite pervasive. Most CMC and virtual classroom systems, for example, support parts of this process, such as information gathering, brainstorming, and collaborative commenting. Some systems, notably ENFI [BPB93], are designed with the goal of creating of a *writing community*. Hypermedia systems, which are described below, are also used for such purposes (called *authoring systems*), especially brainstorming, collaborative commenting. Most of existing collaborative writing systems, for instance, PREP [NKCM90, NCK⁺92], SASE [BNPM93], WE [SWF87], are intended for professional rather than student writers. It is unclear whether the two are different and, if so, what their differences are.

CLARE was not designed to explicitly support collaborative writing. Nevertheless, certain aspects of collaborative writing can benefit from the CLARE approach. First, RESRA can be used as a framework for brainstorming, in particular, when writing is based on the reading conducted in CLARE. Second, a set of related RESRA tuples may serve as an advanced, non-linear *outline* for a new research paper. Learners can even compare these tuples with the ones specified in selected canonical forms to determine whether necessary features and relationships are present. One main characteristic of such an outline is that it defines the *deep-level* rather than *presentational* structure of the paper, as in the traditional outline. Despite this potential, CLARE needs to be extended at the computational level to provide explicit support for collaborative writing tasks.

7.3.3 Hypermedia systems

Hypermedia systems represent an important category of collaborative learning environments. The combination of multi-media, dynamic linking capabilities, and the distributed nature of such a system makes it a powerful tool for:

- Presenting and sharing information;
- Navigating and browsing a complex network of nodes and links; and
- Flexible and collaborative annotation and commenting.

This section reviews three important hypermedia systems/projects: Intermedia, NoteCards, and CoVis. Since there are few empirical studies done on these systems, the focus will be on describing major features of these systems and their support for collaborative learning.

Intermedia. Intermedia, developed at Brown University, is perhaps one of the largest and oldest hypermedia systems designed specifically to support learning. It provides a number of commonly-used tools, such as text editor, graphic editor, timeline editor, and 3-D object viewer. Together, they allow authors to create links to documents of various media. Intermedia is used for a number of learning purposes. Instructors and teachers use Intermedia as an instructional delivery mechanism by organizing and presenting their lecture materials online. Students browse such networks by using Intermedia's built-in graphical browser. More importantly, students can add their own notes and annotations to existing networks of online artifacts. Since all users have access to notes created by other users, they can collaboratively comment on one another's notes. Over years, a number of courses have been taught using Intermedia. However, very few empirical reports on the system can be found in the published literature.

NoteCards. NoteCards, developed at Xerox PARC, is perhaps one of the most widely acclaimed hypermedia systems [Hal87, Tri88, TI87, MI89]. The system was originally intended to support tasks related to design of information systems, such as information gathering and organization. It was later extended to provide multi-user features, such as allowing more than one user working on the same notefile at the same time. Although NoteCards is not a true collaborative learning system, its generic and flexible design allows it to be easily instantiated to support specific learning tasks. Most reports on NoteCards thus far are case studies on the use of this system for various tasks, such as writing, brainstorming.

CoVis Project. CoVis (Collaborative Visualization), a recent project initialized at Northwestern University, aims at providing a distributed multi-media learning environments (DMLE) that support *learning-in-doing* [Pea93]. The proposed system is intended to integrate high-speed networks (ISDN), multi-media, and scientific visualization technology into a media-rich environment that allows students from dispersed locations to engage in authentic science projects with teachers and practicing scientists. CoVis is similar to other hypermedia systems in that they are *infrastructure* technology that can be used to support a wide variety of learning activities. However, CoVis differs from the above systems in that it emphasizes on *multi-media* rather than *hyper-media*. In that sense, CoVis is not truly a hypermedia system.

Hypermedia systems and CLARE. Despite its hypertext-based data model and its support for non-linear navigation, CLARE is not a hypermedia system. Unlike most hypermedia systems which focus on the presentation of information, CLARE emphasizes *representation*, in particular, the process by which such a representation is derived. However, CLARE can benefit from hypermedia systems by incorporating certain interface features from them, such as graphical visualization of network structures, link creation through di-

rect manipulation. Section 8.3 provides several specific proposals on how CLARE might be extended in this direction.

7.3.4 Collaborative knowledge-building tools

One other type of collaborative learning systems that are of increasingly importance is called *collaborative knowledge-building tools*. Compared to other CSCL systems described above, these system possess the following characteristics:

- *Learning is knowledge-building*. Unlike many other CSCL systems that view learning as consisting of such activities as information sharing, reading, writing, the design of knowledge-building systems is based on the view that learning is knowledge-building, and that learning is inherently a collaborative activity. The primary purpose of these systems is to help learners make sense of existing knowledge and construct new knowledge.
- *Integration of technology and pedagogy*. Most of these systems are grounded in one or more established learning theories.
- *Familiar technologies*. Most of these systems provide such capabilities as shared databases, distributed, asynchronous access, hypertext-based navigation, notification, automated activity logs and sometimes, graphical interface.

CSILE, which is described below, is a general-purpose collaborative knowledge-building system based on the theory of *intentional learning*. CLARE is another example of such systems; it is based on the assimilation theory of cognitive learning, and provides specific representational and process-level support for collaborative learning from scientific text.

System description. CSILE (*Computer supported intentional learning Environments*) is an integrated learning system developed at the Ontario Institute for Studies in Education at University of Toronto. At a software level, CSILE consists of a shared or communal database to which all students have access via a local area network. Students create text or graphical notes using built-in editors. They comment on each other's notes and search the database of notes using keywords, author, and other attributes. When a student create a note, he or she associates that note to one of the four predefined types called *thinking types*: *I know*, *high-level questions*, *plan*, and *problem*. When a note is commented on by other student, the note author is notified. These functionalities, however, are hardly unique to CSILE; they are also available in many virtual classroom systems. What differentiates CSILE from other learning systems is its integration of software and a learning approach that grows out of over a decade of research on intentional learning, knowledge-telling, and transformation in writing [BS87]. CSILE represents a joint effort by cognitive scientists, computer scientists, teachers, and students.

Empirical findings. CSILE has been used at primary, elementary, and graduate school levels. Preliminary results from these studies show that CSILE users consistently outperform their non-CSILE counterparts in a number of areas:

- Standardized test scores in reading and language;
- Depth of explanation and knowledge quality in student writing;
- Comprehension of difficult text and transfer of learning to novel problems;
- Identifying knowledge gaps;
- Collaboration among students; and

- Beliefs about learning consistent with a progressive view of knowledge advancement.

Empirical data also demonstrates that, in most of the above areas, each additional year students spend working with CSILE yield additional advantage in results.

CSILE and CLARE. CLARE is similar to CSILE in that they are based on the constructionist paradigm on learning and aim at providing an environment conducive to collaborative construction of knowledge instead of merely information sharing. They both provide detailed, automatic tracking data about the learner's behavior. However, the two are fundamentally different in their approaches:

- *Explicit representation support.* CLARE provides an explicit representation language (RESRA) that serves as a meta-cognitive framework for collaborative learning, while CSILE's four thinking types (*I know, high-level questions, plan, and problem*) allow learners to categorize their intentions.
- *Explicit process-level support.* CLARE defines a process model called SECAI which dichotomizes collaborative learning into two distinct phases: *private* and *public*. CSILE, however, does not provide any process-level guidance.
- *Learning from scientific text.* CLARE is designed to support a specific type of learning — learning from scientific text. CSILE is a more general-purpose environment and, as a result, provides less task-specific services.

One major area in which CLARE can benefit from CSILE's experience is longitudinal studies for assessing specific impact of the system on students' learning outcomes, as measured by quantitative tests and qualitative evaluation. Section 8.3 identifies a few specific directions in which future empirical work on CLARE might be extended.

7.4 Summary

One key feature that distinguishes CLARE from other collaborative learning systems is its theory-driven approach: CLARE's treatment of learning from scientific text as collaborative knowledge construction is based on the constructionist view of science and learning. CLARE's emphasis on the meta-cognitive structure and its role in human learning is guided by the the assimilation theory of cognitive learning, which posits that learning *how to learn* (meta-learning) is more important than learning *what it is* (content learning). This theoretical proposition has been supported by empirical findings on concept maps — one of the two meta-cognitive tools proposed by learning theorists — which show that concept maps are effective in promoting meaningful learning and long-term retention. CLARE extends concept maps and Vee diagrams by proposing a new set of meta-cognitive primitives and canonical forms for characterizing the thematic structure of scientific text and learning activities that are centered on them.

Chapter 8

Conclusions

This chapter concludes the dissertation — the first step of the CLARE journey. It attempts to provide a retrospective view of what has been done thus far and a prospective view of what is still ahead. It begins by summarizing the current work by providing a RESRA representation of the major thematic features of this dissertation. Next, it highlights the main contributions of CLARE to the emerging field of computer-supported collaborative learning. And finally, it identifies a number of directions — both short and long-term — in which the representation, implementation, and experimentation of CLARE might be extended.

8.1 A RESRA representation of this dissertation

CLARE concerns collaborative learning from scientific text, to which this dissertation unquestionably belongs. Hence, it is only fair to conclude this work by applying the principles proposed herein to itself. Below is a summarization of the major themes of the current research expressed in terms of RESRA. The relationships between these nodes are depicted in Figure 8.1:

- CLARE: the approach, the system, and the empirical results ([source](#)): D. Wan: *CLARE: a computer-based collaborative learning environment based on the thematic structure of scientific text*. Ph.D. Dissertation, University of Hawaii, Interdisciplinary Program in Communication and Information Sciences, 1993.

- Limitations of access-oriented CSCL systems (**problem**): Most existing CSCL systems are either access-oriented (e.g., virtual classroom systems such as CoSy at Open University [Mas89]), or media-oriented (e.g., hypermedia systems such as Intermedia at Brown University [Lan90]). Although these systems are found effective in overcoming the geographical, temporal, and media constraints of traditional face-to-face interactions, they do not provide support for explicit, fine-grained representation of the thematic structure of learning artifacts, electronic or printed. The lack of such representation is in part responsible for such problems as *information overload* in virtual classroom systems and *lost-in-the-hyperspace* in hypermedia systems.
- CLARE's approach to collaborative learning (**method**): CLARE is a new approach to collaborative learning based on the assimilation theory of cognitive learning and the content of scientific text. It defines a particular type of learning called *collaborative learning from scientific text* that treats research literature as a basis for knowledge construction. Learning in this context requires learners to reconstruct the conceptual structure of research papers, to uncover inconsistencies, gaps, and other clues for new inquiry, to collaboratively deliberate reasoning behind each learner's positions, and to connect together similar points of view to form a coherent group knowledge base. CLARE comprises three components:
 - A thematically-oriented representation language called RESRA;
 - A collaborative learning model called SECAI; and
 - A distributed computational environment that provides:
 - * Integrated support for RESRA and SECAI;
 - * Hypertext-based interface to scientific text; and
 - * Fine-grained, unobtrusive instrumentation of the learner's usage behavior.
- RESRA (**concept**): RESRA stands for *REpresentational Schema of Research Artifacts*. It is a representational language for characterizing the thematic structure of scientific text and guiding collaborative knowledge construction among learners. It has two main components:

node and link primitives for describing individual thematic features, and canonical forms (CRFs) for capturing artifact-level structures of scientific text.

- SECAI learning model (**concept**): SECAI represents the abbreviations of five key activities in collaborative learning from scientific text: *Summarization, Evaluation, Comparison, Argumentation, and Integration*. These activities are organized into two phases: *exploration* and *consolidation*. Exploration encompasses the first two activities and is performed privately by individual learners. Consolidation consists of the remaining three activities. It involves direct interactions among learners mediated through CLARE.
- CLARE represents a viable approach to support collaborative learning (**claim**): CLARE is a viable environment for supporting collaborative learning from scientific text. The viability of CLARE is based on the viability of its individual components, such as RESRA and SECAI, and the environment as a whole.
- RESRA is a useful representational basis for guiding collaborative learning (**claim**): RESRA is a useful language for mapping essential features of scientific text. It also provides structural model for guiding collaborative learning activities, such as evaluation, comparison, argumentation, and integration of different interpretations and points of view held by individual learners.
- Evaluation experiments on CLARE (**method**): The evaluation of CLARE consists of five sets of experiments involving 24 students from two different computer science classes (one undergraduate and one graduate). Five research papers in software engineering were used in these experiments. The subjects represent a convenient instead of a random sample.
- Outcome, process, and assessment data from the CLARE experiments (**evidence**): The CLARE experiments resulted in a total of 16 group databases that contain about 1,800 nodes and 400 kilobytes of learner-created text. The process data consists of about 80,000 timestamps gathered during the CLARE evaluation. The assessment data consists of 64 post-session questionnaires.

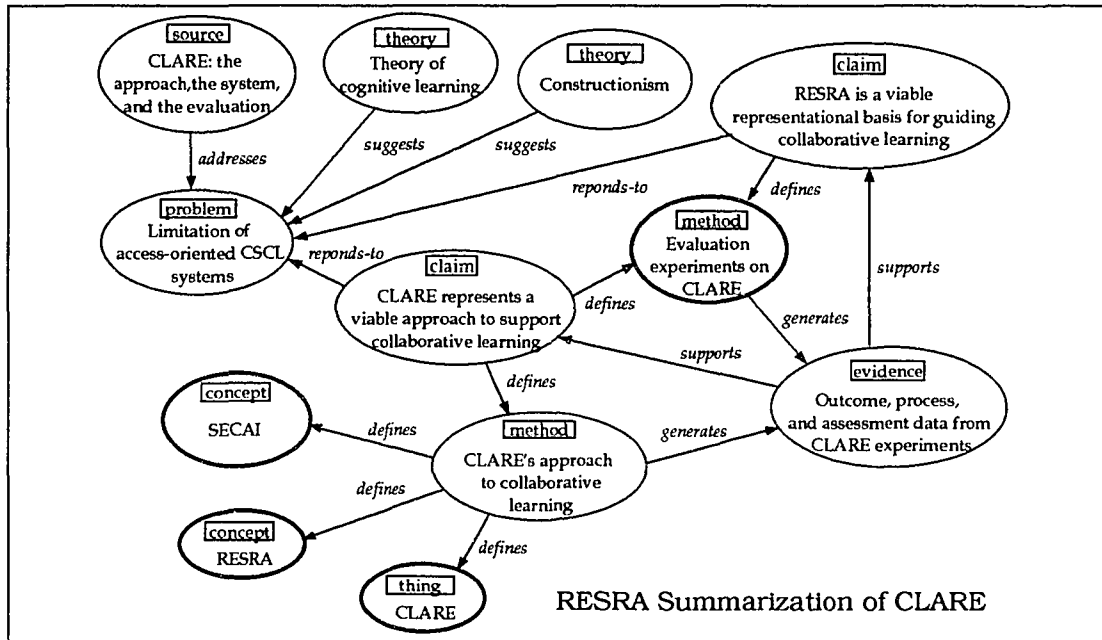


Figure 8.1: A RESRA representation of the major themes of this dissertation

This dissertation belongs to the CRF category of *concept paper*, for it “involves a new conception of a problem, or a new method, technique, or approach to solving an existing problem, or often, both.” The extensive empirical component of this research is treated as the **evidence** in support of this approach.

It should be noted that the focus of the above representation is on the artifact-level major themes. Its purpose is to serve as an overview of the entire work. Individual chapters have their own major themes and corresponding RESRA representation but they are omitted here.

8.2 Main Contributions

In sum, CLARE has made the following four major contributions to the field of computer-supported collaborative learning:

1. It defines a new type of collaborative learning called *learning from scientific text* which links knowledge-building in the scientific community and knowledge-building in the classroom setting via collaborative interpretation and evaluation of the thematic feature of scientific text.
2. It introduces a new knowledge representation language called RESRA that is based on the thematic structure of scientific text, and that provides a structural model for evaluation, comparison, deliberation, and integration of different interpretations of scientific text.
3. It provides a theory-based, distributed collaborative learning environment called CLARE that integrates SECAI, RESRA, an instrumentation mechanism, and a hypertext-based interface.
4. It describes evaluation experiments that provide useful empirical insights on the learner's behavior in using the RESRA representation and the system. They also provide a rich data source for guiding further development of CLARE and future experimentation on collaborative learning in general.

The subsequent sections elaborate upon each of these contributions.

8.2.1 Collaborative learning from scientific text

Learning and research have traditionally been regarded as two quite distinct activities: one concerns the production of new knowledge and the other, the acquisition or transmission of existing knowledge. This view, of course, has been challenged by constructionism, which views that learning, like scientific research, is also knowledge-building. The contribution of CLARE at this level is twofold. First, CLARE makes an explicit attempt to bridge the gap between these two types of knowledge-building via scientific text. In CLARE, scientific text is not merely a primary source of content-level knowledge but also a primary source of meta-knowledge. CLARE encourages learners to discover the

process of scientific knowledge-building by systematically analyzing and evaluating the thematic structures (both intra- and inter-artifact) of research literature. By doing so, for example, learners can come to know how researchers evaluate their peer's work, how they engage in constructive and scholarly argumentation, and so on. They are also encouraged to apply these principles to their own knowledge-building practice, both as students and as researchers.

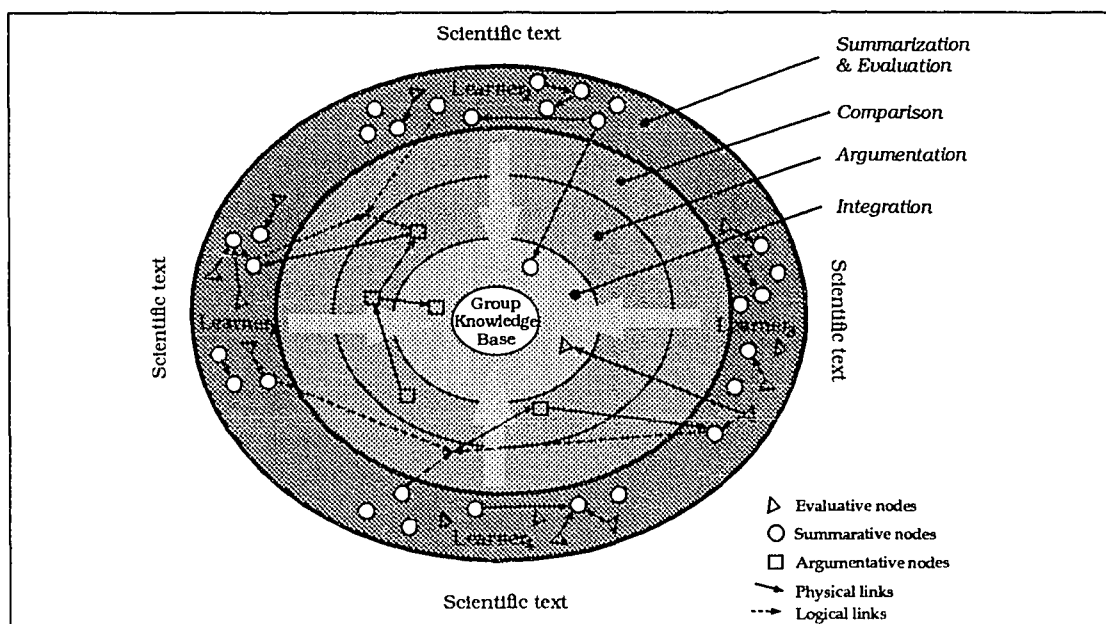


Figure 8.2: CLARE as an environment for supporting collaborative knowledge-building

Second, CLARE defines what collaborative learning from scientific text is by providing an *explicit* process model called SECAI, which specifies the five key learning activities and the relationships between them. Moreover, it also provides explicit representational and computational support for this process.

As depicted in Figure 8.2, the experimental usage of CLARE indicates that learners spent most of their time (66%) on the outmost layer — *summarization/evaluation*, and a decreasing amount of time in the inner layers, shown by the decreasing grey level toward the center. This usage pattern indicates that representing the thematic content of scientific

text is an important but also time-consuming step in the SECAI process. It is also an indication that the goal of collaboratively building a group knowledge-base is a difficult one, and much further research still needs to be done.

8.2.2 RESRA representation language

RESRA is the conceptual basis on which CLARE is built. It provides the essential glue that ties together different components of the system. More importantly, it serves as the meta-cognitive framework that guides learners in their interpretations of scientific text, and their interactions with each other. Although the use of the semi-structured representation to help organize ill-structured task is nothing new, (For a survey, see [LL91]), RESRA has the following unique features:

- It is designed specifically to facilitate collaborative learning by providing a shared frame of reference;
- It provides a fine-grained means for characterizing the important contents of scientific text;
- The layered design of RESRA parallels to that of the SECAI learning model, which gives RESRA additional flexibility and understandability;
- The canonical RESRA forms (CRFs) provides a means to characterize common artifact-level thematic structures of scientific text; and
- RESRA is an open language which can be extended by both the designer and the learner.

Despite a wide variety of mis-interpretations and incorrect usages, the novelty and the usefulness of RESRA has been clearly demonstrated by the usage experience of CLARE. For example, 80% of the learners indicated that RESRA node primitives are very or extremely useful. Certain features of RESRA such as CRFs are still under-utilized.

8.2.3 Design and implementation of the CLARE system

CLARE is a medium-sized, distributed collaborative learning system that is based on the SECAI learning model and the RESRA language. The novelty and contribution of this system reside in the following features:

- *CLARE is grounded in a well-established learning theory.* The design of the system is based on the theory of cognitive learning that is successfully applied to traditional classroom settings over the past three decades [NG84]. One main benefit of such an approach is the consistency and comparability it permits with existing practice and other systems that share the same theoretic principles. For example, since concept mapping and CLARE are both based on cognitive learning theory, they can co-exist in a given learning context, and the learning outcomes from these two processes can be compared.
- *CLARE is an evaluable system.* CLARE was designed to support rigorous empirical experimentation on collaborative learning. To this end, it provides a built-in, fine-grained instrumentation mechanism that unobtrusively keeps track of such information as functions the user invokes, and the time and sequence in which these functions are invoked. During the CLARE experiments, for example, over 80,000 timestamps were collected. Such process-level data is instrumental in understanding the detailed behavior of the users during their interactions with the system.
- *CLARE is an extensible system.* Learning is an ill-structured problem domain, and collaborative learning is particularly so because of group dynamics involved. To accommodate these requirements, CLARE adopts a combination of a layered architecture and an object-oriented design which, along with the flexibility of the Emacs editing environment, makes CLARE an extensible system.

8.2.4 Empirical evaluation of CLARE

Five sets of experiments were conducted as part of the CLARE evaluation. In general, these experiments resulted in primary data about the potential of this new technology and new group process, and shed light on the strengths and weaknesses of the approach. Specifically, these experiments provide evidence in support of the following claims:

- CLARE is a viable tool for supporting collaborative learning from scientific text;
- CLARE provides a useful means of allowing learners to *objectify* both the content and the process of learning from scientific text;
- RESRA is useful in highlighting different points of view among learners;
- RESRA primitives are found useful for *mapping* the thematic features of research literature; and
- The SECAI learning model facilitates the formation of individual views on a research artifact.

The experiments also reveal a number of problems about the RESRA representation, the CLARE system, and collaborative learning in general:

- RESRA is subject to many interpretations;
- *Major themes* of a research paper were missed but its *minor themes* were represented;
- The *major themes* of a research paper were often missed by entire groups of learners;
- CRFs were used by only a fraction of learners;
- The CLARE interface, especially the link mode, is still *less-than-intuitive* to the novice user; and

- Collaborative learning with CLARE is time-consuming.

CLARE is still in an early stage of evolution. Hence, the lessons learned from the current experiments are of particular importance, for they form a basis on which future work will be performed. Uncovering the above problems is an important part of the contribution of this research. The next section presents several potential ways of addressing the above mentioned issues and a number of new directions for further exploration.

8.3 Where CLARE is heading

This dissertation has raised more questions than it has answered. In many ways, it represents only a small step toward a new paradigm of computer-supported collaborative learning. Some basic questions it has raised are: why does it seem so difficult to many learners to *map* the content of an artifact to a representation such as RESRA, as evidenced from the CLARE experiments? Where is the bottleneck: comprehension, understanding of RESRA, analysis, synthesis, articulation, or any combination of these factors? Does the use of RESRA truly enhance the understanding of scientific text? If so, how? If not, why? To answer these and many other similar questions require a more refined RESRA, a more robust CLARE, and additional experimentation. The purpose of this section is to suggest several ways in which RESRA and CLARE can be enhanced, and more rigorous experiments can be performed. It begins by identifying some short-term goals in these areas, followed by a number of long-term directions.

8.3.1 Short-term goals

This section identifies a number of immediate enhancements to CLARE. Most of these extensions are direct response to the findings from the evaluation experiments. The section is organized into three parts: RESRA, CLARE, and experimentation.

RESRA

At the RESRA level, the evaluation indicates that the major problem learners encountered is related to the interpretation of the primitives. The following measures are aimed primarily at alleviating this problem:

- *Categorization of common representation errors*: This will be a continuation of the work that has already started in Section 6.4.2. The focus will be on consolidating various types of representation errors into a list of generalized error types with representative examples. These incorrect usage examples will be made available in CLARE as part of online examples.
- *Guidelines on the usage of RESRA*: The four main areas of confusion on the use of RESRA are:
 1. Distinction between the *major themes* and *minor themes*;
 2. Granularity of representation;
 3. Distinction between *learners' views* and the *authors' views* during summarization; and
 4. Distinction between connotative and denotative interpretations of RESRA primitives.

Specific guidelines will be provided to detail how each of the above situations be handled. Examples derived from the actual database will also be supplied.

- *Explicit support for representation-level deliberation*: A new slot, called *learning type*, will be added to all existing RESRA node primitives. This slot will initially take only two values: *content* (default) or *meta*. For example, with this extension, a learner can make a claim about a RESRA primitive by creating a claim node, and

then set the *learning type* slot to *meta*. CLARE will provide querying and comparison capabilities on this slot, for example, “list all meta-level question nodes created by user X.”

- *Refinement of online examples*: Real examples of RESRA instances offer a concrete way of illustrating what a good RESRA instance (for instance, problem) is like. The current examples were not rated as adequate by the learners during the experiment. New and better examples will need to be introduced.
- *Broadening and refining the current set of CRFs*: The CRF is still an under-utilized feature of CLARE. Its usefulness may become more evident when learners begin to focus their attentions on the *major themes* of an artifact. The existing CRFs, however, need to be extended to incorporate more artifact types, such as *case studies*.

CLARE

The following features are incremental extensions to the current version of CLARE; they do not require redesign or major restructuring of the current system.

- *Enhancement in the reliability and robustness of CLARE*: One major step in this direction is to upgrade CLARE to use the most recent versions of EGRET and database server (HBS), both of which have improved reliability and performance.
- *Improvement to the CLARE interface*: A short-term solution to the CLARE interface problem is to incorporate an auxiliary graphical browser that possesses the following functionalities:
 - Link creation by direct selection of the source and destination nodes;
 - Typed icons for predefined RESRA node primitives;
 - Node selection, deletion, locking, unlocking by direct manipulation.

This browser will be used in conjunction with the existing CLARE's buffer-based interface.

- *Enhancement to the comparison mode:* The assumption behind the current CLARE comparison mode is (1) the number of nodes created by each learner is small; and (2) learners focus on the representation of the major themes of an artifact. Since the experiment reveals that learners often represent many minor themes and create a relatively large number of nodes, the comparison mode will need to be extended to provide following capabilities:
 - To differentiate major themes from minor ones;
 - For major themes, comparison are made at the artifact level; and
 - For minor themes, comparison are made at the semantic unit level if the number of nodes exceeds a user definable threshold value, and at the artifact level otherwise.
- *Node retyping and merging:* To allow learners to change the type of a node, or merge two nodes into a single node, for example, during link creation. A further enhancement will be to make these capabilities available through the browser.
- *Search capabilities:* The initial implementation might restrict search only to the Subject field instead of the entire node content.

Experimentation

There were a few limitations in the design and execution of the CLARE evaluation experiments:

- *Absence of pre- and post-session tests.* No tests on individual learning styles or locus of control were done prior to the experiment. Nor were comprehension tests done after the session.

- *Group assignment.* The use of existing project groups in the first set of experiments seems to have created certain bias on the result.
- *Improper selection of research papers.* For example, the first paper used in the experiment was considered as too long.

The following recommendations will help lead to better CLARE experimentation:

- *Explicitly defined independent and dependent variables:* These variables are based on the hypotheses to be tested. Example independent variables are: summarization strategies and the number of passes over the source nodes. An example dependent variable is the instructor-assigned quality rating on the learning artifact generated.
- *Careful selection of artifacts:* The content, style, length, and type of artifacts used need to be carefully weighted according to the learner's background.
- *True experimental design:* Learners are divided into experimental groups (CLARE users) and control groups (non-CLARE users), Group members are randomly assigned. Pre-tests, such as on learning styles, cognitive styles, locus of control, are conducted when necessary. Learning outcomes are measured quantitatively (e.g., instructor assigned ratings on the quality of the artifact produced), and qualitatively, such as learners' perceived usefulness of CLARE features.
- *Training and pilot testing:* Pilot testing is done on learner groups of the same characteristics as the intended subjects. Both off-line and online training is provided to the first time CLARE users.
- *Demographic data:* Such data can help answer questions such as whether or not there is any difference between male and female learners in terms of usage strategies and learning outcomes.

8.3.2 Long-term directions

This section identifies a number of long-term research directions for CLARE. Like the previous section, it is organized into three parts: RESRA, CLARE, and experimentation.

RESRA

Domain-specific RESRA (DSR). RESRA is a generic representation language that can be used in various subject domains, ranging from software engineering to organizational behavior. The utility of RESRA can be extended by creating domain-specific instantiation of this representation:

- A set of RESRA node instances, such as “Software quality crisis” (`problem`), “Formal technical review (FTR)” (`method`), “FTR improves software quality” (`claim`), and link instances that express the relationships between these nodes, such as `FTR improves software quality` $\xrightarrow{\text{responds-to}}$ `software quality crisis`; and
- A set of CRFs that characterize the exemplary structure of research artifacts in that domain, for example, “experience reports in software engineering.”

These DSR instances pertain to a particular subject matter, such as “software engineering.” They might be generated by previous CLARE users or experts on the subject, such as the course instructor. They may be linked to the original source artifact.

DSR shares the same objectives as the general-purpose RESRA: facilitating interpretations of scientific text and collaborative deliberation of these interpretations. Moreover, it has one advantage: because of its domain-specificity, it provides learners more concrete guidance on how to use the representation to interpret the content of scientific text. DSR serves the following specific roles in collaborative learning:

- *As an index to the core knowledge of the chosen domain:* This role is especially evident when DSR instances are created by experts in the field.
- *As a seed knowledge-base:* To new comers of a field or new students in a course, such a collection of DSR instances represent a starting point for exploring other research artifacts in the domain.

Like the generic RESRA, DSR is dynamic: new instances can be added when necessary, either by designated individuals or any CLARE users. In fact, it is much simpler to extend DSR than to extend the generic RESRA primitives.

RESRA case libraries. RESRA case libraries (RCL) refer to collections of actual RESRA instances. A *case* is defined as a *complete* representation of an artifact by a given learner. It contains not only that outcome but also the process steps that lead to the outcome. RCL is a generalization of RESRA examples. It differs from examples in two ways:

- Examples are often given as individual nodes or tuples, while RCL is defined at the level of artifacts and learners; and
- Examples represent only the outcome, but RCL encompasses both outcome and the process data.

RCL is also different from domain-specific RESRA (DSR). DSR is RESRA-centered. It selects node and tuple instances based on their domain significance. In contrast, RCL is learner-centered and much less selective in what to be included.

RCL is significant for three reasons. First, it supports the situated nature of RESRA — the correctness and soundness of RESRA interpretation and usage are assessed within the context of the learner and the artifact being represented. Second, it provides a learner-centered high-level construct for viewing and analyzing learner’s interpretation of RESRA

and the content of artifacts. When supported with necessary indexing and query capabilities, a learner can ask CLARE to show “all existing representations of artifact X that contain at least two but no more than five tuples and, that are ranked as *good* in quality.” Third, when supported with animation features of CLARE (see below), a learner can *re-play* the process by which a given representation is derived.

CLARE

Advanced interface support. As shown in Section 6.3, CLARE interface is the most important barrier in the current implementation. The previous section offers a short-term solution by incorporating a graphical browser that supplements the current buffer-based interface. In a long run, CLARE needs to move toward a complete graphical interface, similar to systems such as NoteCards [HMT87]. Furthermore, to realize the representational potentials of RESRA requires support for automatic graph layout, visualization, and animation. For example, with visualization capabilities, learners will be able see the overall structure of an artifact, and the structure of each learner’s representation. They may also be able to superimpose two or more learners’ representations to discern differences and similarities between them, or to zoom in to a cluster of tuples to have a closer view of what they are. Moreover, such tools can also be used to visualize where in the artifact are the *information-rich spots*, as measured by the number of summarative nodes originated from them, and where are the *center of controversy*, or clusters of evaluative nodes. Similarly, learners can use animation tools to *replay* the sequence of CLARE commands that lead to the creation of a given node, tuple, or case.

Inferencing capabilities. CLARE is a knowledge-based system; it embeds a knowledge representation language (RESRA) and structural knowledge about selected scientific text (CRFs). It currently provides a simple *advise* feature that, based on the system’s knowledge about the current artifact, the corresponding CRF definition, and what the learner has done

so far, suggests to the learner what node and/or tuple to consider next. This feature may be extended to support inferencing capabilities. For example, if a learner attempts to create a link between a `method` and `problem`, and the corresponding CRF contains the tuple `method` *generates* `evidence`, CLARE may suggest to the learner to consider changing the `problem` to `evidence`. Similarly, CLARE may send a warning message to a learner if, based on the analysis of its sentence structures, a `question` node he just created does not seem like a question.

Integration with other learning environments CLARE currently supports one particular type of learning — learning from scientific text. There are also many other types of learning, two of which are listed below:

- *Project-based learning*: A form of *learning-by-doing* that is commonly found in science classes, for example, doing a physics or chemistry experiment. Systems such as CoVis [Pea93] provide explicit support for such learning;
- *Intentional learning*: A type of writing-oriented learning with particular emphasis on the deliberation of the reasoning behind each position the learner takes. CSILE is an environment for supporting intentional learning [SB93].

A learning setting such as a college-level biology class typically involves a combination of several types of learning. Based upon the current state of technology, however, it will require learners to use several systems at the same time, which is often neither economically feasible, nor pedagogically productive. Therefore, integrated learning environments, which can bring together systems such as CLARE and CSILE, are called for.

Experimentation

Longitudinal field studies. At an empirical level, one major challenge facing CLARE is to find institutional settings in which longitudinal studies (similar to those done by the CSILE team [SB93]) can be carried out. Such studies, which may typically last for a semester or longer, will be conducted as part of normal requirements of the selected courses. Research artifacts to be used will be selected directly from the course reading list. Experiment and control groups will be employed so that differences in their learning outcomes, as measured by their understanding of the materials and the quality of the artifacts produced, can be assessed. The results from these studies will show the effect of CLARE on the learning process and outcome under the intended usage setting.

Comparative studies. Under the above experimental settings, a number of control methods can be used to assess the impact of different independent variables:

- *Type of research artifacts:* Good candidates include conceptual, opinion, and empirical papers.
- *Subject domain:* The subject matter of learning may range from literature to computer science and zoology.
- *Learner backgrounds:* Selecting learners from different academic majors: science, engineering, social sciences, humanities, and so on.
- *Expert-novice users:* Compare expert CLARE users with the ones who are the first-time users to determine whether different strategies are used.

In addition, experiments may also be conducted to compare CLARE with other learning methods, such as concept mapping. The problem, however, is that these methods are not always comparable. For example, concept maps are rarely used to support collaborative learning. On the other hand, RESRA/CLARE was designed primarily for such a purpose.

Hence, to compare them empirically requires one to *scale-down* RESRA by omitting its evaluative primitives, or *scale-up* concept maps to equivalent functionality of RESRA. Fortunately, the former is relatively easy to do.

Appendix A

CLARE Assessment Questionnaire

1. CLARE provides you a useful means of characterizing what is important in the paper.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)

2. CLARE allows you to represent your view of the paper at a fine-grained level without sacrificing the comparability with the views of your peers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)

3. CLARE allows you to see the differences between your view of the paper and those of your peers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)

4. CLARE enables you to integrate your view of the paper with those of your peers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)

5. The predefined templates capture the thematic structures of example research papers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)

6. CLARE templates have led you to form a view of the paper close to those of your peers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
_____	_____	_____	_____	_____

7. CLARE's two-step (i.e., exploration and consolidation) process model allows you to form your own view of the paper without potential distractions from those of others.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
_____	_____	_____	_____	_____

8. CLARE's two-step (i.e., exploration and consolidation) process model gives each person fair opportunity to do his or her part.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please specify)
_____	_____	_____	_____	_____

9. You're becoming better at evaluating research papers because, through CLARE, you learn how your peers approach the same problem.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
_____	_____	_____	_____	_____

10. You're becoming better at evaluating research papers because, over time, you become more knowledgeable about the subject domain.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
_____	_____	_____	_____	_____

11. It is *useful* to be able to read other people's views on the content of the paper.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
_____	_____	_____	_____	_____

12. CLARE's comparison mode is useful because it enables you to see the differences between your view of the paper and those of your peers.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
<input type="checkbox"/>				<input type="checkbox"/>

13. CLARE is useful in helping you understand the content of the paper in the way that is not possible before.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
<input type="checkbox"/>				<input type="checkbox"/>

14. CLARE is useful in helping you understand your peers' perspectives in the way that is not possible before.

Strongly Disagree	Disagree	Agree	Strongly Agree	Other (please explain)
<input type="checkbox"/>				<input type="checkbox"/>

15. During the CLARE session just completed, how many times did you find that key points in the paper *cannot* be represented using the predefined *node* primitives?

Never	1-2 times	3-5 times	6 or more times
<input type="checkbox"/>			

16. During the CLARE session just completed, how many times did you find that important *relationships* between different key points in the paper *cannot* be represented using the predefined *link* primitives?

Never	1-2 times	3-5 times	6 or more times
<input type="checkbox"/>			

17. During the CLARE session just completed, how many times did you find CLARE templates sensitizing you to ambiguities in the paper which you would have not noticed otherwise?

Never	1-2 times	3-5 times	6 or more times
<input type="checkbox"/>			

18. How often do you think CLARE should be used in classroom settings?

At least once a month	At least once a semester	At least once at school	Not used

19. Please rank the following CLARE features in terms of usefulness.

Scale: 5 = extremely useful; 0 = not useful at all.

- *Node* primitives
- *Link* primitives
- Templates
- Two-phase process model
- Comparison mode
- Online examples
- Other (please specify)

20. Please rank the following factors as obstacles to using CLARE.

Scale: 5 = greatest obstacle, 0 = not an obstacle at all.

- Interface
- *Node* primitives
- *Link* primitives
- Link mode
- Two-phase process model
- Other (please specify)

21. Please enter below whatever comments and suggestions you might have about your experience with CLARE:

Appendix B

CLARE Feedback Questionnaire

To help improve the quality of CLARE, we'd like to hear from you about your experience with the system. Please take a few minutes and answer the following questions. Write down any additional comments you might have in blank regions (use the reverse side if necessary). Your cooperation is appreciated.

1. Have you ever used any other collaborative learning systems, such as electronic bulletin board systems, electronic meeting-rooms, group brainstorming tools, hypermedia systems? If so, please describe your experience in relation to CLARE.

2. Please rate each of the following on a scale of 1 to 5, where 1 indicates *Low Quality* and 5 indicates *High Quality*:
 - (a) Quality of the CLARE User's Guide: 1 2 3 4 5
 - (b) Quality of the CLARE system: 1 2 3 4 5
 - (c) Quality of the CLARE learning process: 1 2 3 4 5
 - (d) Quality of the CLARE learning outcome: 1 2 3 4 5

3. What were the biggest problems you had in learning to use CLARE? Please briefly list them below:

4. If you could make any change to CLARE to make it more effective and easier to use as a collaborative learning tool, what would those changes be?

5. On a scale of 1-5, where 1 is *Extremely Dissatisfied* and 5 is *Extremely Satisfied*, please rate your feelings about CLARE: 1 2 3 4 5

6. Suppose CLARE is available for public use, would you recommend using it for studying research papers?

Appendix C

A CLARE Session Summary

Artifact: R.F. Flynn and M. Dorfman *The Automated Requirements Traceability System (ARTS): An Experience of Eight Years*¹.

Participants:

- Todd
- Mary
- Scott
- Chris

Total number of nodes created: 92

- Summmarative nodes: 58
- Evaluative nodes: 0
- Argumentative nodes: 34
- Integrative links: 10

C.1 Scott

Claim442

- Subj: Basic Analysis Phase Issues

¹This appendix contains a linearized version of the CLARE database used by the case described in Section 6.6. It is automatically generated by CLARE hardcopy generator.

- Desc.: It is implied that the "Waterfall Model" is a widely used model in software development. The authors claim that in general, the most difficult problem lies in accurately defining the requirements. This causes a problem of perception which is further compounded by the fact that requirements are increased and modified during the process of development. This claim underlines the need for requirements standards. It is mentioned later that the analysis phase is often glossed over and that the errors that creep in during this phase can cause serious problems during operations. At the operations stage, it is both time consuming and difficult to correct those problems, which can otherwise be easily corrected in the analysis phase. Hence the authors believe that it is useful in the long run to focus more effort on the analysis stage.

• Exploration:

- *responds-to* \rightarrow *problem434(Traceabil)*
- *is-alternative-to* \rightarrow *claim452(OnEarly)*
- *strengthens* \rightarrow *theory456(Theorybe)*
- *summarizes* \rightarrow *Introduction(Introduct)*

• Deliberation:

- *evaluates* \leftarrow *critique636 (requireme)*

Claim452

- Subj: On Early Traceability Implementations
- Desc.: Traceability has been a manual rather than an automated process, making it unreliable, among other shortcomings. Many commercial packages do not support traceability and comparison is difficult since traceability standards do not exist.
- Exploration:
 - $\xrightarrow{\text{suggests}}$ *problem434(Traceabil)*
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*
 - $\xleftarrow{\text{is-alternative-to}}$ *claim442 (Basic Ana)*

Theory456

- Subj: Theory behind ARTS
- Desc.: An automated database containing all requirements with cross references, with the capability of storing and retrieving these in tree formats would improve the quality of work at the requirements stage and hence lead to the creation of a better system.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*
 - $\xleftarrow{\text{strengthens}}$ *claim442 (Basic Ana)*
 - $\xleftarrow{\text{strengthens}}$ *claim538 (Current S)*

Thing458

- Subj: Overview of ARTS
- Desc.: "ARTS is a database management system". The database stores requirements in the form of plain text in records and ARTS keeps track of these. It has many advantages, but among the disadvantages is the lack of "analytical capability".

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*

Method460

- Subj: Features of ARTS
- Desc.: ARTS is written in Fortran and is designed to run on many different computer systems. The core piece of software is the database manager. Traceability components were added after a database was chosen. One of the special features of ARTS is the ability having multiple entries in a single field. ARTS was developed by a succession of cycles of prototyping and development.

- Exploration:

- $\xrightarrow{\text{generates}}$ *evidence554(Shortcomi)*

- $\xrightarrow{\text{summarizes}}$
ARTSdescription(ARTSdesc)

- Deliberation:

- $\xleftarrow{\text{augments}}$ *suggestion666 (Improveme)*

Claim528

- Subj: ARTS is Flexible
- Desc.: ARTS accomplishes the a basic objective for such systems, namely "flexibility".

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *UseofARTS(UseofAR)*

- $\xleftarrow{\text{presupposes}}$ *claim574 (Authors')*

- Deliberation:

- $\xleftarrow{\text{augments}}$ *suggestion638 (flexible)*

- Integration:
 - (Mary)^{comprises}_← claim418 (ARTS and)

Claim538

- Subj: Current Status of ARTS
- Desc.: ARTS is probably the best of the few available tools that perform requirements management. It has evolved into a powerful user-friendly tool with excellent technical support. It is very flexible and certain functionalities can be transferred to other platforms. It has become popular enough for several organizations to request access to it. The use of ARTS has help bring down costs, improve quality and reduce project duration, in addition to providing good management capability.
- Exploration:
 - ^{strengthens}_→ theory456(Theorybe)
 - ^{summarizes}_→ ARTStoday()
- Deliberation:
 - ^{challenges}_← question618 (Can ARTS)
- Integration:
 - ^{is-related-to}_→ claim518(successo)
 - (Mary)^{is-similar-to}_← claim590 (ARTS toda)(Todd)

Evidence554

- Subj: Shortcomings of ARTS

- Desc.: Despite its advantages, ARTS has many shortcomings, some of which are listed here. It can be slow when the databases are huge. Many screens do not have graphical capabilities to show tree structures. Modification of the database is not possible when there are multiple users, which is a significant shortcoming. ARTS is not fully automated, despite what the name suggests.

- Exploration:
 - ^{summarizes}_→ Difficulties(Difficult)
 - ^{generates}_← method460 (Features)

Claim560

- Subj: Some Practical Limits
- Desc.: Flexibility is a necessary characteristic of requirement management systems. Too much flexibility in a system though, could lead to improper implementation. Besides this, tools can never be fully automated.
- Exploration:
 - ^{presupposes}_→ claim574(Author's')
 - ^{summarizes}_→ Conclusions(Conclusio)
- Deliberation:
 - ^{challenges}_← question640 (Flexibili)
 - ^{evaluates}_← critique658 (Too much)
- Integration:
 - ^{comprises}_→ claim596(Conclusio)(Todd)

Thing570

- Subj: Internal Structure
- Desc.: "ARTS is a software system that is made up of a set of progChriss and procedures" along with several databases, system libraries and user interfaces. The internal structure of ARTS is based on a "top-down" hierarchical approach. Its use typically begins during the analysis phase of the project.
- Exploration:

- $\xrightarrow{\text{summarizes}}$ *UseofARTS(UseofAR)*

Other524

- Subj: Enhancements of ARTS
- Desc.: A lot of effort was put into making ARTS user-friendly. The user interface upgrades to support different formats for input and extraction of data. The VMS system is the standard operating system for running ARTS. The developers tried to implement ARTS on other platforms, but due to various problems, they preferred the DEC VAX system.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Evolution()*

- Deliberation:

- $\xleftarrow{\text{augments}}$ *suggestion732 (Developme)*

Problem434

- Subj: Traceability Issues
- Desc.: The authors imply that though traceability is a necessity for any good requirements managements system, it has met with

very limited success, due to reasons that are not explained well. The main source of this problem may lie in what has been mentioned in the last lines of the abstract: the lack of standards for stating requirements. The author's approach to the problem is briefly described and it is stated that the success of traceability systems is proportional to the ease of being adapted to different "application environments".

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Abstract()*

- $\xleftarrow{\text{responds-to}}$ *claim574 (Authors')*

- $\xleftarrow{\text{suggests}}$ *claim452 (On Early)*

- $\xleftarrow{\text{responds-to}}$ *claim442 (Basic Ana)*

- Deliberation:

- $\xleftarrow{\text{evaluates}}$ *critique616 (over stat)*

- $\xleftarrow{\text{evaluates}}$ *critique678 (About Tra)*

- Integration:

- $\xrightarrow{\text{comprises}}$
problem424(Requireme)(Chris)

- $\xrightarrow{\text{comprises}}$
problem466(Requireme)(Todd)

Claim574

- Subj: Authors' Conclusions
- Desc.: Software tools created for such purposes must be easily adaptable to varying application environments. ARTS has this property and is considered a trailblazer in this field.

- Exploration:
 - $\xrightarrow{\text{responds-to}}$ *problem434(Traceabil)*
 - $\xrightarrow{\text{presupposes}}$ *claim528(ARTSisF)*
 - $\xrightarrow{\text{summarizes}}$ *Abstract()*
 - $\xleftarrow{\text{presupposes}}$ *claim560 (Some Prac)*

- Deliberation:
 - $\xleftarrow{\text{evaluates}}$ *critique642 (ARTS' ada)*
 - $\xleftarrow{\text{evaluates}}$ *critique614 (incomplet)*
-

Question692

- Subj: Re: Requirements management and traceability
- Desc.: It seems rather surprising that the issue of traceability is not understood properly, despite its obvious importance. The authors have not clearly explained why this issue has been misunderstood and has received little attention. One would have expected the opposite: a great deal of effort would be put into studying this field.
- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *problem424(Requireme)(Chris)*

Suggestion694

- Subj: Re: Requirements analysis tools
- Desc.: I think that it is very important to introduce formal standards for RM and traceability. I feel that this would in itself solve some of the problems that tools like ARTS attempt to solve.

- Deliberation:
 - $\xrightarrow{\text{augments}}$ *problem438(Requireme)(Chris)*

Suggestion696

- Subj: Re: problem with ARTS
- Desc.: With any complex software application, there will probably be some end-users who will use the system in a manner that may be incorrect, but more convenient for them; hence the requirement that the systems must be flexible enough to give correct responses.

- Deliberation:
 - $\xrightarrow{\text{augments}}$ *problem542(problemw)(Mary)*

Question710

- Subj: Re: Life Cycle Model
- Desc.: It would seem that the "Waterfall" model is the most popular model used. Since many problems originate from the lack of emphasis placed on the requirements analysis phase, I feel that some of these problems could be eliminated if the "Waterfall" model were to be replaced by some other model which has inherently more emphasis on requirements. The "Incremental" model is probably one such alternative.

- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *theory502(LifeCycl)(Chris)*

Question714

- Subj: Re: success of ARTS
- Desc.: There are very few tools of the genre of ARTS. Coupled with the fact that there are no standards in the field, making qualitative comparisons is a little difficult. Making quantitative comparisons is extremely difficult, if not impossible at this point in time. I feel that if there were a way to assign some numerical values to various modes of these tools, it will be a more reliable way of evaluating their performance. This though will not happen till formal standards are established.
- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *claim518(successo)(Mary)*

Suggestion718

- Subj: Re: ARTS today - a viable tool
- Desc.: The authors state that a complete overhaul of ARTS is being considered. Given the growing popularity of Unix and other platforms, along with rapid technological changes, it makes sense to move away from VAX systems alone. Fortran is not as commonly used today, as compared with the time when ARTS was created. I feel that to remain viable, ARTS will have to be made portable to commonly used platforms. The fact that screens with graphical capabilities are becoming increasingly common can only work to the advantage of ARTS.
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *claim590(ART Stoda)(Todd)*

C.2 Chris

Problem424

- Subj: Requirements management and traceability
- Desc.: The authors of this paper emphasize the fact that requirements management and traceability are extremely important for the development of large systems, but are topics that are not well understood and require further study.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Abstract()*
 - $\xleftarrow{\text{suggests}}$ *theory502 (Life Cycl)*
- Deliberation:
 - $\xleftarrow{\text{challenges}}$ *question692 (Re: Requi)*
- Integration:
 - $\xleftarrow{\text{comprises}}$ *(Mary)* *problem434*
(Traceabil)(Scott)

Problem436

- Subj: Inadequacy of traceability methods
- Desc.: The inadequacy of the conventional traceability methods, which required manual linking of requirement paragraph numbers using large tables, has been pointed out.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*
 - $\xleftarrow{\text{responds-to}}$ *claim454 (A databas)*
- Deliberation:

- $\overset{\text{evaluates}}{\leftarrow}$ critique742 (conventio)

- Integration:

- (Mary) $\overset{\text{is-same-perspective-as}}{\leftarrow}$ problem412
(need of t)

Problem438

- Subj: Requirements analysis tools

- Desc.: Although there are a few tools that support requirements analysis, the lack of support for requirements traceability among some of these tools is a problem that needs to be addressed. The absence of traceability standards is also a problem that needs to be studied.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ Introduction(Introduct)
- $\overset{\text{responds-to}}{\leftarrow}$ claim544 (ARTS -One)

- Deliberation:

- $\overset{\text{augments}}{\leftarrow}$ suggestion694 (Re: Requi)

Theory444

- Subj: Steps in requirements engineering process

- Desc.: LMSC's System Engineering Organization has identified 3 steps in the ideal requirements engineering process: 1. identification 2. allocation 3. demand

- Exploration:

- $\overset{\text{defines}}{\rightarrow}$ concept496(Allocation)
- $\overset{\text{defines}}{\rightarrow}$ concept494(Identific)
- $\overset{\text{defines}}{\rightarrow}$ concept498(Flowdown)
- $\overset{\text{summarizes}}{\rightarrow}$ UseofARTS(UseofAR)

Problem446

- Subj: Project failure and its connection to poor requirements analysis

- Desc.: Poorly performed requirements analysis has been identified as one of the main reasons for the failure of many large systems.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ Introduction(Introduct)
- $\overset{\text{suggests}}{\leftarrow}$ theory502 (Life Cycl)

- Integration:

- (Mary) $\overset{\text{comprises}}{\leftarrow}$ problem410 (requireme)

Claim454

- Subj: A database tool can significantly improve requirements management

- Desc.: The designers of the ARTS system felt that a database tool which can store and retrieve the different requirements and their linkages in tree formats, would significantly improve the requirements process .

- Exploration:

- $\overset{\text{responds-to}}{\rightarrow}$ problem436(Inadequac)
- $\overset{\text{summarizes}}{\rightarrow}$ Introduction(Introduct)

- Deliberation:

- $\overset{\text{evaluates}}{\leftarrow}$ critique644 (tree form)
- $\overset{\text{evaluates}}{\leftarrow}$ critique728 (requireme)

Claim468

- Subj: Tools have to be flexible
- Desc.: The authors claim that for a requirements tool to be successful, it has to be flexible.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Conclusions(Conclusio)*
 - $\xleftarrow{\text{supports}}$ *evidence470 (Lack of f)*

Evidence470

- Subj: Lack of flexibility in tools leads to a number of difficulties in their use
- Desc.: The authors provide evidence to show how methodology complications arise when a tool does not fit the function for which it was designed. As an example, they point out how many users have trouble accepting a unique REQID for each requirement. The lack of flexibility implies that users cannot use paragraph numbers which are preferred in some applications. Flexibility would make the tool easier to use and make it better suited for different applications.
- Exploration:
 - $\xrightarrow{\text{supports}}$ *claim468(Toolshav)*
 - $\xrightarrow{\text{summarizes}}$ *Difficulties(Difficult)*
 - $\xleftarrow{\text{generates}}$ *method506 (Tool-to-t)*

Concept482

- Subj: Requirements Analysis
- Desc.: It is the process through which the exact requirements of a system are identified. It is usually the first step that is performed in the design of any software system.

- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*

Concept488

- Subj: Traceability
- Desc.: Identification and documentation of the derivation path (upward) and allocation/flowdown path (downward) of requirements in the hierarchy.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*

Concept494

- Subj: Identification
- Desc.: This process involves identifying the different requirements of the system.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *UseofARTS(UseofAR)*
 - $\xleftarrow{\text{defines}}$ *theory444 (Steps in)*

Concept496

- Subj: Allocation
- Desc.: The process of partitioning, budgeting or apportioning a performance or functional requirement to the physical subelements of a system.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *UseofARTS(UseofAR)*
 - $\xleftarrow{\text{defines}}$ *theory444 (Steps in)*
- Deliberation:
 - $\xleftarrow{\text{challenges}}$ *question730 (Hierarchi)*

Concept498

- Subj: Flowdown
- Desc.: The process of generating lower level requirements from allocated higher level requirements.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ Use of ARTS (Use of AR)
 - $\xleftarrow{\text{defines}}$ theory444 (Steps in)

Concept500

- Subj: Requirement
- Desc.: An authorized and documented need of a customer.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ Introduction (Introduce)
- Deliberation:
 - $\xleftarrow{\text{challenges}}$ question756 (concept ?)

Theory502

- Subj: Life Cycle Model
- Desc.: The system lifecycle model breaks up a project into several discrete phases. The phases in the model include requirements analysis, design, building, testing, delivery and maintenance of the system.
- Exploration:
 - $\xrightarrow{\text{suggests}}$ problem424 (Requirement)
 - $\xrightarrow{\text{suggests}}$ problem446 (Project f)
 - $\xrightarrow{\text{summarizes}}$ Introduction (Introduce)

- Deliberation:

- $\xleftarrow{\text{challenges}}$ question710 (Re: Life)

Method506

- Subj: Tool-to-task paradigm
- Desc.: The tool to task paradigm is used to analyze the difficulties associated with the development and application environments. These problems need to be completely understood before any tool can be built. This paradigm has four components: situation, computer, people and data. This paradigm attempts to analyze the characteristics of the application environment associated with the requirements management tools.
- Exploration:
 - $\xrightarrow{\text{generates}}$ evidence470 (Lackoff)
 - $\xrightarrow{\text{summarizes}}$ Tool considerations (Toolcons)

Claim544

- Subj: ARTS -One of first tool to perform requirements mgt
- Desc.: The authors claim that ARTS is one of the first tools to perform that requirements management function along with traceability.
- Exploration:
 - $\xrightarrow{\text{responds-to}}$ problem438 (Requirement)
 - $\xrightarrow{\text{summarizes}}$ ARTS today()
- Deliberation:
 - $\xleftarrow{\text{augments}}$ suggestion646 (type of c)

Suggestion632

- Subj: Problem with ARTS
- Desc.: Requirements management tools can be improved by understanding the tools' application environment, and by designing the tools to be flexible enough so that they can adapt to any application's environment .
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *problem542(problemw)(Mary)*

Critique658

- Subj: Too much flexibility
- Desc.: I agree that too much flexibility may not always be desirable. Besides, I feel that tools that are tailored for a particular application's needs (although not flexible enough for other applications) may be much better and efficient than those that are very flexible.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim560(SomePrac)(Scott)*

Evidence662

- Subj: ARTS is not all that flexible
- Desc.: Please see Evidence "Lack of flexibility in tools leads to a number of difficulties" which shows that some features in ARTS were not all that flexible.
- Deliberation:
 - $\xrightarrow{\text{counters}}$ *claim522(reasonsb)(Mary)*

Suggestion666

- Subj: Improvements to ARTS
- Desc.: There are a number of areas in which improvements can be made. The entire tools may have to be rewritten either in C or object oriented C or in some fourth generation language. There is a need to incorporate distributed databases into the tool. ARTS was originally designed for DEC VAX computers. Although versions of ARTS were created for other platforms like CDC Cyber, IBM mainframes and Univac 1100s they were not very successful. However for the success of ARTS there is need for development of versions for these machines. Graphical output format may have to be supported in future versions.
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *method460(Features)(Scott)*

Critique678

- Subj: About Traceability issues
- Desc.: The reason why traceability has met with limited issues is mainly because there are no good requirements management tools that address this issue. Traditionally this has been accomplished by manual methods which were inadequate and error-prone for large systems.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *problem434(Traceabil)(Scott)*

Question618

- Subj: Can ARTS be transferred to other platforms?

- Desc.: It is not clear what you mean by 'certain functionalities can be transferred to other platforms'. It is mentioned in the paper that the versions developed for CDC Cyber, IBM mainframe and others were not very successful

- Deliberation:

- $\xrightarrow{\text{challenges}}$ *claim538(CurrentS)(Scott)*

Suggestion680

- Subj: ARTS and traceability
- Desc.: A further improvement to the traceability abilities of ARTS can be achieved by using hypertext techniques .

- Deliberation:

- $\xrightarrow{\text{augments}}$ *claim418(ARTSand)(Mary)*

Critique690

- Subj: Are problems in sys reqmt management due to life cycle model ?
- Desc.: In this problem, it is mentioned that problems in system requirement management arise due to 'inadequacies in the methodology followed (life cycle model)' — It is not clear what inadequacies in life cycle model really affect the requirements management function....

- Deliberation:

- $\xrightarrow{\text{evaluates}}$ *problem532(Majorpro)(Todd)*

C.3 Mary

Problem410

- Subj: requirements inadequacies
- Desc.: A software development project starts with analysis of the problem which results in a set of requirements. Setting exact requirements is difficult and moreover requirements themselves keep changing with time. These necessitates proper management of requirements. Inadequate requirements results in software, poor in terms of quality and maintainability.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*

- $\xleftarrow{\text{responds-to}}$ *claim418 (ARTS and)*

- Integration:

- $\xrightarrow{\text{comprises}}$ *problem446(Projectf)(Chris)*

Problem412

- Subj: need of traceability
- Desc.: Requirements should be traceable. Accomplishing traceability manually is impractical and most tools available for analysis do not support traceability.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*

- $\xleftarrow{\text{responds-to}}$ *claim418 (ARTS and)*

- Integration:

- $\xrightarrow{\text{is-same-perspective-as}}$ *problem436(Inadequac)(Chris)*

Claim418

- Subj: ARTS and traceability
- Desc.: ARTS is a DBMS into which requirements can be stored & maintained. ARTS perform bookkeeping, rather than analysis. Requirements are stored as text or table or figure in a tree structure, which helps in ARTS' traceability capability.
- Exploration:
 - $\overset{\text{responds-to}}{\rightarrow}$ *problem410(requireme)*
 - $\overset{\text{responds-to}}{\rightarrow}$ *problem412(needoft)*
 - $\overset{\text{defines}}{\rightarrow}$ *concept550(descripti)*
 - $\overset{\text{summarizes}}{\rightarrow}$ *Introduction(Introduct)*
- Deliberation:
 - $\overset{\text{augments}}{\leftarrow}$ *suggestion648 (bookkeepi)*
 - $\overset{\text{augments}}{\leftarrow}$ *suggestion680 (ARTS and)*
- Integration:
 - $\overset{\text{comprises}}{\rightarrow}$ *claim528(ARTSisF)(Scott)*

Other440

- Subj: Evolution of ARTS
- Desc.: ARTS mainly runs on DEC VAX. The user interface, particularly the data entry part and add requirements part, has been improved. Formats of report generated have also been enhanced. Other enhancements are internal performance improvements, consistency and completeness checks, etc. Other developments had been in terms of documentation, training people, user support and configuration management.

- Exploration:
 - $\overset{\text{summarizes}}{\rightarrow}$ *Evolution()*

Claim518

- Subj: success of ARTS
- Desc.: ARTS is a popular requirement management software. It has opportunity to be marketed. It still fully supported and it is used throughout entire system development lifecycle.
- Exploration:
 - $\overset{\text{summarizes}}{\rightarrow}$ *ARTStoday()*
- Deliberation:
 - $\overset{\text{challenges}}{\leftarrow}$ *question714 (Re: succe)*
- Integration:
 - $\overset{\text{is-related-to}}{\leftarrow}$ *claim538 (Current S)(Scott)*
 - $\overset{\text{comprises}}{\leftarrow}$ *claim590 (ARTS toda)(Todd)*

Claim522

- Subj: reasons behind success
- Desc.: ARTS provides good requirement management techniques including traceability, visibility and control. The system is flexible and extensible. Input is convenient and storage format is modifiable. It is easy to use and modifiable. ARTS has stood the test of time.
- Exploration:

- $\xrightarrow{\text{summarizes}}$ ARTStoday()

• Deliberation:

- $\xrightarrow{\text{presupposes}}$ claim650(successo)

- $\xleftarrow{\text{counters}}$ evidence662 (ARTS is n)

Problem542

• Subj: problem with ARTS

• Desc.: There are software complications (in terms of performance, complexity, report, generation, multiple users, etc.), application complications when users use it in a non-standard fashion (top-down approach not followed, traceability ignored, etc.), methodology complications when ARTS is used for some other purpose, people related complications, etc.

• Exploration:

- $\xrightarrow{\text{summarizes}}$ Difficulties(Difficult)

• Deliberation:

- $\xleftarrow{\text{augments}}$ suggestion632 (Problem w)

- $\xleftarrow{\text{augments}}$ suggestion696 (Re: probl)

• Integration:

- $\xrightarrow{\text{is-similar-to}}$
problem594(Difficult)(Todd)

Concept550

• Subj: description of ARTS

• Desc.: ARTS mainly consists of a database management system and traceability components. But it can be used for entire development project. The DBMS is internally developed. A requirement object is a record which includes the following fields : - an id - id of the parent(s) - source paragraph no. (for top level req.) - allocations (sub-requirements) - req. level - a pointer to the description of the req. - date, etc. Select, save, manipulate, report generate, etc. are allowed. ARTS was developed through rapid prototyping and incremental development using Fortran language.

• Exploration:

- $\xrightarrow{\text{summarizes}}$ ARTSsystem(ARTSsyst)

- $\xleftarrow{\text{defines}}$ claim418 (ARTS and)

Method492

• Subj: how it works, how to use

• Desc.: The five progChris in ARTS are : database progChris, text processor, specification input system, external forms, text checker. Requirements originate with customer. ProgChris management passes it to system engineers who perform requirement management. Top level requirements are first defined. Lower level requirements in the hierarchy are created and traceability maintained. This top down approach consists of identification, allocation and flowdown. Data entry personnel physically inputs requirements in ARTS.

• Exploration:

- $\xrightarrow{\text{summarizes}}$ UseofARTS(UseofAR)

Problem562

- Subj: difficulty with tools
- Desc.: difficulty with tools may be categorised into: situation - internal & external factors affecting the task. people - developers, users, management, customers. data - requirements. computer - hardware & software (os, ui, dbms, etc.).

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Toolconsiderations(Toolcons)*

Theory572

- Subj: successful tool
- Desc.: for a tool to be successful, it must be flexible in terms of checking, verification, multiple processing, data manipulation, configuration, database structure, user interface, output. Support must come from management, developers, application support group and users (feedback). The tool must span the entire life cycle.
- Exploration:

- $\xrightarrow{\text{summarizes}}$ *Conclusions(Conclusio)*

Concept576

- Subj: future directions
- Desc.: to make a requirement engineering tool successful, requirement standards are to be defined, fragmentation eliminated and allocation & flowdown defined. Current trends are on using AI techniques, language processors, hypertext techniques, oodbms, database servers across networks. The tools are to be

written in C, C++, 4gl, or in embedded structured query languages. They must run on wide variety of machines.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ *ARTSystem(ARTSyst)*

Critique614

- Subj: incomplete description
- Desc.: "Software tools created for such purposes" - what purposes is not explained.
- Deliberation:

- $\xrightarrow{\text{evaluates}}$ *claim574(Authors')(Scott)*

Critique616

- Subj: over statement
- Desc.: the authors do not say that success of traceability systems is "proportional" to their adaptability.
- Deliberation:

- $\xrightarrow{\text{evaluates}}$ *problem434(Traceabil)(Scott)*

Critique636

- Subj: requirement problems
- Desc.: the authors highlight the inadequacy of the analysis phase. They are talking about a problem that exists and which is well known. This cannot be a claim.
- Deliberation:

- $\xrightarrow{\text{evaluates}}$ *claim442(BasicAna)(Scott)*

Suggestion638

- Subj: flexible to a certain extent
- Desc.: The source says that ARTS can be used in a bookkeeping mode. So "flexibility" is in respect to that only. In fact if used in some other way, ARTS can be quite rigid. The section 'Difficulties with use of ARTS' speaks of such situations. For instance, ARTS cannot handle situations where requirements are not flowed down in a top-down fashion, resulting in too many derivations.
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *claim528(ARTSisF)(Scott)*

Question640

- Subj: Flexibility
- Desc.: Based on what criteria can one decide that a system is flexible ?
- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *claim560(SomePrac)(Scott)*

Critique642

- Subj: ARTS' adaptability
- Desc.: ARTS is NOT adaptable to varying application environments. It mainly runs on DEC VAX.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim574(Authors')(Scott)*

Critique644

- Subj: tree formats
- Desc.: This is more a theory, based on which ARTS system was developed.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim454(Adatabas)(Chris)*

Suggestion646

- Subj: type of claim
- Desc.: type of claim : fact
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *claim544(ARTS - One)(Chris)*

Suggestion648

- Subj: bookkeeping
- Desc.: "ARTS perform bookkeeping, rather than analysis." - What is meant by this is : ARTS cannot automate the process of analysis.
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *claim418(ARTSand)*

Claim650

- Subj: success of ARTS II
- Desc.: please refer to claim "success of ARTS"
- Deliberation:
 - $\xrightarrow{\text{presupposes}}$ *claim522 (reasons b)*

Critique654

- Subj: methodology
- Desc.: "The author describes the methodology ..." - as it is clearly stated, the node is a method node and not a claim node.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim530(Largesca)(Todd)*

Critique668

- Subj: requirements ?
- Desc.: requirements cannot be claim.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim548(Needfor)(Todd)*

Critique742

- Subj: conventional traceability methods
- Desc.: conventional traceability methods were not inadequate. They are "labor-intensive, prone to errors, and provided little visibility to the process."
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *problem436(Inadequac)(Chris)*
- **Suggestion750**
- Subj: methodology complications
- Desc.: methodology complications also contribute to difficulties and limitations with arts
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *problem594(Difficult)(Todd)*

Feedback754

- Subj: feedback754
- Desc.: there was a problem node and a critique (to the problem) node, both of which were on display. i wanted to endorse the critique, but the problem got endorsed (may be because of some emacs problem). There was no way i could delete the endorsement. there is no way to delete an integration also.

Question756

- Subj: concept ?
- Desc.: the node provides the definition of "requirement". Even if one wants to call it a concept, it is too well known and too obvious to include here.
- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *concept500(Requireme)(Chris)*

Critique758

- Subj: evolution
- Desc.: the node describes the evolution of ARTS. It is NOT a method. Methods are procedures or techniques used for generating evidence of a claim.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *method564(Developme)(Todd)*

Critique760

- Subj: description
- Desc.: The node does not describe ARTS and its use

- Deliberation:

- $\xrightarrow{\text{evaluates}}$ thing552(Briefdis)(Todd)

C.4 Todd

Problem466

- Subj: Requirements traceability
- Desc.: Requirements traceability, used in the development of large projects, is generally not performed correctly.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ Abstract()

- Integration:

- (Mary) $\xleftarrow{\text{comprises}}$ problem434
(Traceabil)(Scott)

Method520

- Subj: ARTS paradigm
- Desc.: Flexible tool is needed to fit a particular environment while dealing with Requirements management. ARTS paradigm is based on this consideration.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ Abstract()

Claim530

- Subj: Large scale system development process
- Desc.: The author describes the methodology currently adopted for the development of a large scale system development project.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ Introduction(Introduct)

- Deliberation:

- $\xleftarrow{\text{evaluates}}$ critique652 (critique6)

- $\xleftarrow{\text{evaluates}}$ critique654 (methodolo)

Problem532

- Subj: Major problems faced in the system requirement management

- Desc.: Author identifies these to be the major problems in req. mgmt. The problems arise due to :i)Inadequacies in the methodology followed(life-cycle model in this case) and ii) The limitations of the existing tools.

- Exploration:

- $\xrightarrow{\text{summarizes}}$ Introduction(Introduct)

- Deliberation:

- $\xleftarrow{\text{evaluates}}$ critique690 (Are probl)

Claim548

- Subj: Need for developing ARTS

- Desc.: Author indicates the requirement which led to the development of ARTS

- Exploration:

- $\xrightarrow{\text{summarizes}}$ Introduction(Introduct)

- Deliberation:

- $\xleftarrow{\text{evaluates}}$ critique668 (requireme)

Thing552

- Subj: Brief discription and use of ARTS
- Desc.: Author's description of the system - specifying its capabilities.
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Introduction(Introduct)*
- Deliberation:
 - $\xleftarrow{\text{evaluates}}$ *critique760 (descripti)*

Thing558

- Subj: General discription of ARTS
- Desc.: The development of ARTS started as a research project to build req. traceability system for software engrs. A contemporary requirement from Lockheed for formalizing the set of concepts and procedures for Req Engg. function led to a close coupling of the functional requirement and traceability in this project. ARTS became a system engg. tool. The development base for ARTS was limited by the hardware and software standards existing at the time. The database used for the purpose is designed to make the tool flexible for the user(and cortrespondingly for the dev. environment).
- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *ARTSdescription(ARTSdesc)*

Method564

- Subj: Development methodology for ARTS

- Desc.: The development methodology for the ARTS software was based on a combination of rapid prototyping and in- cremental development. This approach partially satisfied the user(Lockheed) requirements who could then utilize the system and eventually lead to in- cremental dev. Once the initial problems were overcome, the system was released to the general user community.The prototype became an established tool. The system has been in- crementally developed since then without any major redesign.

The ARTS software was written in the Fortran progChris- ming language.

- Exploration:
 - $\xrightarrow{\text{summarizes}}$ *ARTSdescription(ARTSdesc)*
 - $\xleftarrow{\text{summarizes}}$ *other508 ()*
- Deliberation:
 - $\xleftarrow{\text{evaluates}}$ *critique758 (evolution)*

• Other578

- Subj: Evolution of ARTS
- Desc.: Evolution of ARTS implied incorporating enhancments and modifications and doing standardization. These can be characterized into: 1) Feature addition/modification 2) Interface enhancement/modification - the emphasis is to evolve an easy method of entering data. 3) Improved reports/output formats. 4) Internal performance improvements, consistency and completeness checks. 5) Other features eg. online news, status monitoring etc. A PC version of the software was later released which provided a subset of ARTS capability.
Documentation of ARTS was given due im- portance.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *Evolution()*

Thing428

- Subj: ARTS system & interface
- Desc.: The interface provides user access to text editor, database, spec. input system, forms utility and text checker.
- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *UseofARTS(UseofAR)*

Method582

- Subj: Use of ARTS to specify the system req.
- Desc.: The use of ARTS begins during the analysis or requirements definition phase of a project. A potential user first collects all the requirements for the projects. They originate with the customer who interfaces with progChris management. A Software engineer handles the systems requirements management tasks. Data entry or application support personnel performs the physical task of adding or manipulating the data.

Once the top-level or system requirements are collected and a system hierarchy is defined, the ARTS database record storage format is created for the requirements information. An ARTS database is initialized, the storage format is added, and the requirements are entered individually, in batch groups, or from output from the specification input progChris. These requirements are allocated to lower-level components in the system hierarchy and lower-level requirements are created and added to the database as needed, while maintaining traceability linkages.

The requirements engineering process supported by ARTS consists of the steps of identification, allocation, and flowdown (a top-down approach). Once the system hierarchy has been defined, system requirements are identified first for the top-level "block." After being identified, they are added to the ARTS database in the storage format specified. These system requirements are later "allocated" or pointed to one or more appropriate lower-level blocks through the use of an ALLOCATION field entry. After the appropriate analysis, these requirements are then "decomposed" or "flowed down" into more detailed requirements for each of the lower-level blocks to which the requirement was allocated.

The higher-level requirements are sometimes referred to as "parents" and the more detailed as "children." The lower-level requirements are connected to their parents through the use of a DERIVATION field that contains the REQIDs or identifiers of the parent requirements. Every lower-level requirement must have at least one parent. ARTS allows multiple parents (limited currently to eight) for a requirement. The lower-level requirements are sometimes referred to as "derived requirements."

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *UseofARTS(UseofAR)*

Method584

- Subj: Following improper req. engg. process for traceability
- Desc.: The system allows users not to follow ideal or recommended approach to req. mgmt. if they so desire. An example of such a

behavior is cited by the author for "after-the-fact" bookkeeping mode. The system allows the traceability links to be specified and reports to be generated.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *UseofARTS(UseofAR)*

Method586

- Subj: Output report generation

- Desc.: Various formats are available for generation of output reports. Several options are available for each report that enable the user to include fields of interest and to vary output sizes.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *UseofARTS(UseofAR)*

Claim590

- Subj: ARTS today - a viable tool

- Desc.: ARTS has evolved to become a useful tool to perform req. mgmt. including traceability. The software is fully supported. User support is available and application support is also provided. There is a real need for req. mgmt. techniques including traceability and these are well handled thru ARTS. The system has useful features e.g. it is flexible and extensible(a feature useful due to the diversity in the req. mgmt. of diff. systems), the system poses no limit to number of requirements, the system is small-enough(modifiability is easy). ARTS has matured as a software product.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *ARTStoday()*

- Deliberation:

- $\overset{\text{augments}}{\leftarrow}$ *suggestion718 (Re: ARTS)*

- Integration:

- $\overset{\text{comprises}}{\rightarrow}$ *claim518(successo)*

- $\overset{\text{is-similar-to}}{\rightarrow}$ *claim538(CurrentS)(Scott)*

Problem594

- Subj: Difficulties and limitations

- Desc.: The complications faced with ARTS may be categorized among the following: a)Software complications - the capability limitations associated with the features may affect an application, obstacles with report generation, speed limitations due to creation of a large requirement database etc. In the normal case (i.e. if the user follows the recommended methodology) then most of these limitations won't come into play. b)Applications complications - when users try to do more from the software than it is designed for. c) People-related complications- due to the impact of personnel on tool acceptance. May be associated to the users own whims and prejudices.

- Exploration:

- $\overset{\text{summarizes}}{\rightarrow}$ *Difficulties(Difficult)*

- Deliberation:

- $\overset{\text{augments}}{\leftarrow}$ *suggestion750 (methodolo)*

- Integration:

- $\overset{\text{is-similar-to}}{\leftarrow}$ *(Mary) problem542 (problem w)*

Claim596

- Subj: Conclusions
 - Desc.: Flexibility is essential for such systems but excess flexibility can lead to incorrect applications of the tool.
 - Exploration:
 - $\xrightarrow{\text{summarizes}}$ *Conclusions(Conclusio)*
 - Integration:
 - *(Mary)* $\xleftarrow{\text{comprises}}$ *claim560* *(Some Prac)(Scott)*
-

Critique728

- Subj: requirement mgmt and tree struct
- Desc.: The author presupposes that in the req. analysis the requirements would generate a tree structure. What if a requirement has multiple parent requirements? Such requirements might naturally arise in real-systems. In such systems the structuring based on tree concept may itself lead to inadequate representation of requirements. This may suggest an inadequacy of the traceability model also.
- Deliberation:
 - $\xrightarrow{\text{evaluates}}$ *claim454(Adatabas)(Chris)*

Question730

- Subj: Hierarchical allocation

- Desc.: The concept of allocation usually implies a tree-type hierarchical decomposition. Allocation is an important concept but wouldn't the type of decomposition implied lead to an incomplete implementation of the concept.
- Deliberation:
 - $\xrightarrow{\text{challenges}}$ *concept496(Allocation)(Chris)*

Suggestion732

- Subj: Development platform
- Desc.: A WINDOWS environment seems to be a useful one and should be attempted.
- Deliberation:
 - $\xrightarrow{\text{augments}}$ *other524(Enhanceme)(Scott)*

C.5 Argumentation nodes

1. *critique760* description (Mary):
2. *critique758* evolution (Mary):
3. *question756* concept ? (Mary):
4. *suggestion750* methodology complications (Mary):
5. *critique742* conventional traceability methods (Mary):
6. *suggestion732* Development platform (Todd):
7. *question730* Hierarchical allocation (Todd):
8. *critique728* requirement mgmt and tree struct (Todd):

9. **suggestion718** Re: ARTS today - a viable tool (Scott):
10. **suggestion696** Re: problem with ARTS (Scott):
11. **question714** Re: success of ARTS (Scott):
12. **suggestion694** Re: Requirements analysis tools (Scott):
13. **question692** Re: Requirements management and traceability (Scott):
14. **critique690** Are problems in sys reqmt management due to life cycle model ? (Chris):
15. **question710** Re: Life Cycle Model (Scott):
16. **suggestion680** ARTS and traceability (Chris):
17. **critique678** About Traceability issues (Chris):
18. **critique668** requirements ? (Mary):
19. **suggestion666** Improvements to ARTS (Chris):
20. **evidence662** ARTS is not all that flexible (Chris):
21. **critique658** Too much flexibility (Chris):
22. **critique654** methodology (Mary):
23. **claim650** success of ARTS II (Mary):
24. **suggestion648** bookkeeping (Mary):
25. **suggestion646** type of claim (Mary):
26. **critique644** tree formats (Mary):
27. **critique642** ARTS' adaptability (Mary):
28. **question640** Flexibility (Mary):
29. **suggestion638** flexible to a certain extent (Mary):
30. **critique636** requirement problems (Mary):
31. **suggestion632** Problem with ARTS (Chris):
32. **question618** Can ARTS be transferred to other platforms? (Chris):
33. **critique616** over statement (Mary):
34. **critique614** incomplete description (Mary):

C.6 RESRA tuples

claim574 $\xrightarrow{\text{presupposes}(\text{Scott})}$ *claim528*
claim538 $\xrightarrow{\text{strengthens}(\text{Scott})}$ *theory456*
method460 $\xrightarrow{\text{generates}(\text{Scott})}$ *evidence554*
claim560 $\xrightarrow{\text{presupposes}(\text{Scott})}$ *claim574*
claim442 $\xrightarrow{\text{strengthens}(\text{Scott})}$ *theory456*
claim442 $\xrightarrow{\text{is-alternative-to}(\text{Scott})}$ *claim452*
claim442 $\xrightarrow{\text{responds-to}(\text{Scott})}$ *problem434*
claim452 $\xrightarrow{\text{suggests}(\text{Scott})}$ *problem434*
claim574 $\xrightarrow{\text{responds-to}(\text{Scott})}$ *problem434*
claim418 $\xrightarrow{\text{defines}(\text{Mary})}$ *concept550*
claim418 $\xrightarrow{\text{responds-to}(\text{Mary})}$ *problem412*
claim544 $\xrightarrow{\text{responds-to}(\text{Chris})}$ *problem438*
theory502 $\xrightarrow{\text{suggests}(\text{Chris})}$ *problem446*
theory502 $\xrightarrow{\text{suggests}(\text{Chris})}$ *problem424*
method506 $\xrightarrow{\text{generates}(\text{Chris})}$ *evidence470*
claim454 $\xrightarrow{\text{responds-to}(\text{Chris})}$ *problem436*

evidence470 $\xrightarrow{\text{supports(Chris)}}$ claim468
 theory444 $\xrightarrow{\text{defines(Chris)}}$ concept498
 theory444 $\xrightarrow{\text{defines(Chris)}}$ concept494
 theory444 $\xrightarrow{\text{defines(Chris)}}$ concept496
 claim418 $\xrightarrow{\text{responds-to(Mary)}}$ problem410

question692* $\xrightarrow{\text{challenges(Scott)}}$ problem424
 suggestion694* $\xrightarrow{\text{augments(Scott)}}$ problem438
 suggestion696* $\xrightarrow{\text{augments(Scott)}}$ problem542
 question710* $\xrightarrow{\text{challenges(Scott)}}$ theory502
 question714* $\xrightarrow{\text{challenges(Scott)}}$ claim518
 suggestion718* $\xrightarrow{\text{augments(Scott)}}$ claim590
 critique616* $\xrightarrow{\text{evaluates(Mary)}}$ problem434
 critique636* $\xrightarrow{\text{evaluates(Mary)}}$ claim442
 suggestion638* $\xrightarrow{\text{augments(Mary)}}$ claim528
 question640* $\xrightarrow{\text{challenges(Mary)}}$ claim560
 critique642* $\xrightarrow{\text{evaluates(Mary)}}$ claim574
 critique644* $\xrightarrow{\text{evaluates(Mary)}}$ claim454
 suggestion646* $\xrightarrow{\text{augments(Mary)}}$ claim544
 suggestion648* $\xrightarrow{\text{augments(Mary)}}$ claim418
 critique614* $\xrightarrow{\text{evaluates(Mary)}}$ claim574
 claim522 $\xrightarrow{\text{presupposes(Mary)}}$ claim650*
 critique652* $\xrightarrow{\text{evaluates(Mary)}}$ claim530
 critique654* $\xrightarrow{\text{evaluates(Mary)}}$ claim530
 critique668* $\xrightarrow{\text{evaluates(Mary)}}$ claim548
 critique742* $\xrightarrow{\text{evaluates(Mary)}}$ problem436
 suggestion750* $\xrightarrow{\text{augments(Mary)}}$ problem594
 question756* $\xrightarrow{\text{challenges(Mary)}}$ concept500
 critique758* $\xrightarrow{\text{evaluates(Mary)}}$ method564

critique760* $\xrightarrow{\text{evaluates(Mary)}}$ thing552
 suggestion632* $\xrightarrow{\text{augments(Chris)}}$ problem542
 critique658* $\xrightarrow{\text{evaluates(Chris)}}$ claim560
 evidence662* $\xrightarrow{\text{counters(Chris)}}$ claim522
 suggestion666* $\xrightarrow{\text{augments(Chris)}}$ method460
 critique678* $\xrightarrow{\text{evaluates(Chris)}}$ problem434
 question618* $\xrightarrow{\text{challenges(Chris)}}$ claim538
 suggestion680* $\xrightarrow{\text{augments(Chris)}}$ claim418
 critique690* $\xrightarrow{\text{evaluates(Chris)}}$ problem532
 critique728* $\xrightarrow{\text{evaluates(Todd)}}$ claim454
 question730* $\xrightarrow{\text{challenges(Todd)}}$ concept496
 suggestion732* $\xrightarrow{\text{augments(Todd)}}$ other524
 claim560 $\xrightarrow{\text{comprises(Mary)}}$ claim596
 claim590 $\xrightarrow{\text{is-similar-to(Mary)}}$ claim538
 claim590 $\xrightarrow{\text{comprises(Mary)}}$ claim518
 claim538 $\xrightarrow{\text{is-related-to(Mary)}}$ claim518
 claim418 $\xrightarrow{\text{comprises(Mary)}}$ claim528
 problem542 $\xrightarrow{\text{is-similar-to(Mary)}}$ problem594
 problem410 $\xrightarrow{\text{comprises(Mary)}}$ problem446
 problem412 $\xrightarrow{\text{is-same-perspective-as(Mary)}}$
 problem436
 problem434 $\xrightarrow{\text{comprises(Mary)}}$ problem466
 problem434 $\xrightarrow{\text{comprises(Mary)}}$ problem424

Table C.1: Summarative nodes (EX) at a glance

<i>User</i>	<i>PR</i>	<i>CL</i>	<i>CO</i>	<i>ME</i>	<i>EV</i>	<i>TH</i>	<i>SO</i>	<i>TI</i>	<i>OT</i>	<i>Total</i>
Scott	1	6	0	1	1	1	0	2	1	13
Chris	4	3	6	1	1	2	0	0	0	17
Mary	4	3	2	1	0	1	0	0	1	12
Todd	3	4	0	5	0	0	0	3	1	16

Table C.2: Evaluative nodes (EX) at a glance

<i>User</i>	<i>Critique</i>	<i>Question</i>	<i>Suggestion</i>	<i>Total</i>
Scott	0	0	0	0
Chris	0	0	0	0
Mary	0	0	0	0
Todd	0	0	0	0

Table C.3: Summarative nodes (CON) at a glance

<i>User</i>	<i>PR</i>	<i>CL</i>	<i>CO</i>	<i>ME</i>	<i>EV</i>	<i>TH</i>	<i>SO</i>	<i>TI</i>	<i>OT</i>	<i>Total</i>
Scott	0	0	0	0	0	0	0	0	0	0
Chris	0	0	0	0	1	0	0	0	0	1
Mary	0	1	0	0	0	0	0	0	0	1
Todd	0	0	0	0	0	0	0	0	0	0

Table C.4: Evaluative nodes (CON) at a glance

<i>User</i>	<i>Critique</i>	<i>Question</i>	<i>Suggestion</i>	<i>Total</i>
Scott	0	3	3	6
Chris	3	1	3	7
Mary	10	2	4	16
Todd	1	1	1	3

Bibliography

- [Abe81] R.P. Abelson. Psychological status of the script concept. *American Psychologist*, 36:715–29, 1981.
- [ADS84] W. Arnaudin, J.J. Dunn, and T.H. Shafer. Concep mapping in college science teaching. *Journal of Research in Science Teaching*, November 1984.
- [AH83] J.W. Alba and L. Hasher. Is memory schematic? *Psychological Bulletin*, 93:203–31, 1983.
- [ANH78] David P. Ausubel, J.D. Novak, and H. Hansesian. *Educational psychology: a cognitive view*. Hold, Rinchart and Winston, New York, 2nd edition, 1978.
- [Aus63] David P. Ausubel. *The psychology of meaningful verbal learning*. Grune & Stratton, 1963.
- [Baz88] Charles Bazerman. *Shaping written knowledge: the genre and activity of the experimental article in science*. University of Wisconsin Press, 1988.
- [BL66] Peter Berger and Thomas Luckman. *The Social Construction of Reality: a Treatise in the Sociology of Knowledge*. Doubleday, 1966.
- [Blo56] Benjamin Bloom, editor. *Taxonomy of educational objectives - handbook 1: cognitive domain*. Longmans, Green and Co., 1956.
- [BNPM93] Ronald M. Baecker, Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby. The user-centred iterative design of collaborative writing software. In *Proceedings of ACM INTERCHI'93 Conference on Human Factors in Computing Systems*, pages 399–405, 1993.
- [BPB93] Bertram Bruce, Joy K. Peyton, and Trent Batson, editors. *Network-based classrooms: promises and realities*. Cambridge University Press, 1993.
- [Bre93] Robert S. Brewer. URN: A new way to think about Usenet. Technical Report CSDL-TR-93-06, University of Hawaii, Department of Information and Computer Sciences, May 1993.

- [BS87] C. Bereiter and M. Scardamalia. *The psychology of written composition*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.
- [Bus90] Marilyn Bush. Improving software quality: The use of formal inspections at the JET Propulsion Laboratory. *1990 IEEE 12th International Conference on Software Engineering*, pages 196–199, March 1990.
- [CB88] J. Conklin and L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, 6(4):303–331, October 1988.
- [Cli90] Joseph W. Cliburn, Jr. Concept maps to promote meaningful learning. *Journal of College Science Teaching*, pages 212–217, February 1990.
- [Com87] Apple Computer. *Human Interface Guidelines: the Apple Desktop Interface*. Addison-Wesley Publishing Co., Reading, MA, 1987.
- [CSGMW92] Violetta Cavalli-Sforza, Gareth Gabrys, Alan M. Lesgold, and Arlene Weiner. Engaging students in scientific activity and scientific controversy. In Kishore Swaminathan, editor, *Proceedings of 1992 AAAI Workshop on Communicating Scientific and Technical Knowledge*, pages 99–106, San Jose, 1992.
- [CSM79] Bill Curtis, Sylvia Sheppard, and Phil Milliman. Third time charm: stronger prediction of programmer performance by software complexity metrics. In *Proceedings of the fourth international conference on software engineering*, pages 356–60, 1979.
- [CY91] E. Jeffrey Conklin and K.C. Burgess Yakemovic. A process-oriented approach to design rationale. *Human-Computer Interaction*, 6(3,4):357–391, 1991.
- [Fag76] Michael E. Fagan. Design and code inspections to reduce errors in program development. *IBM System Journal*, 15(3):182–211, 1976.
- [FD90] R.F. Flynn and M. Dorfman. The automated requirement traceability systems (ARTS): an experience of eight years. In *Proceedings of the fifth international conference on software engineering*, pages 423–438, 1990.
- [Gai92] Brian R. Gaines. Social dynamics of scholarly publication. In *Proceedings of 1992 AAAI Workshop on Communicating Scientific and Technical Knowledge*, 1992.

- [Hal87] Frank G. Halasz. Reflections on Notecards: Seven issues for the next generation of hypermedia systems. In *ACM Hypertext' 87 Proceedings*, pages 345–365, 1987.
- [Har89] Steven Harnad. Minds, machines and Searle. *Journal of Theoretical and Experimental Artificial Intelligence*, 1(1):5–25, 1989.
- [Har91] Steven Harnad. Post-gutenberg galaxy: The fourth revolution in the means of production of knowledge. *Public-Access Computer Systems Review*, 2(1):39–53, 1991.
- [Hil88] S.P. Hiltz. Collaborative learning in a virtual classroom: Highlights of findings. In *Proceedings of ACM CSCW' 88 Conference on Computer-Supported Cooperative Work*, pages 282–290, 1988.
- [HMT87] Frank G. Halasz, Thomas P. Moran, and Randall H. Trigg. Notecards in a nutshell. In *Proceedings of ACM CHI+GI' 87 Conference on Human Factors in Computing Systems and Graphics Interface*, pages 45–52, 1987.
- [Joh91] Philip M. Johnson. The EGRET project: Exploring open, evolutionary, and emergent collaborative systems. In *Proceedings of the 1991 ECSCW Developer's Workshop*, September 1991.
- [Joh94] Philip M. Johnson. Experiences with EGRET: An exploratory group work environment. *Collaborative Computing*, 1(1), 1994.
- [JT93] Philip M. Johnson and Danu Tjahjono. Improving software quality through computer supported collaborative review. In *Proceedings of the Third European Conference on Computer Supported Cooperative Work*, September 1993.
- [JWT⁺93] Philip M. Johnson, Dadong Wan, Danu Tjahjono, Kiran Kavoori, and Robert Brewer. EGRET requirements specification. Technical Report CSDL-TR-93-02, University of Hawaii, Department of Information and Computer Sciences, 1993.
- [KC81] K.D. Knorr-Centina. *The manufacture of knowledge: an essay on the constructivist and contextual nature of science*. Pergamon Press, Oxford, 1981.
- [Knu92] Donald Knuth. Learning from our errors. In C. Floyd, H. Zullighoven, R. Budde, and R. Keil-Slawik, editors, *Software development and reality construction*, pages 28–30. Springer-Verlag, 1992.

- [KR77] W. Kunz and H. Rittel. Issues as elements of information systems. Technical report, Institute of Urban and Regional Development, University of California at Berkeley, 1977.
- [KTBB92] Simon M. Kaplan, William J. Tolone, Douglas P. Bogia, and Celsina Bignoli. Supporting collaborative software development with Conversation-Builder. In *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work*, 1992.
- [Lan90] G.P. Landow. Hypertext and collaborative work: the example of Intermedia. In R.E. Kraut J. Galegher and C. Egido, editors, *Intellectual Teamwork: Social Foundations of Cooperative Work*, pages 407–428. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1990.
- [LL91] Jintae Lee and Kum-Yew Lai. What's in design rationale? *Human-Computer Interaction*, 6(3,4):251–280, 1991.
- [Mas89] Robin Mason. An evaluation of CoSy on an Open University course. In Robin Mason and Anthony Kaye, editors, *Mindweave: communication, computers, and distance education*. Pergamon Press, Oxford, 1989.
- [MI89] Catherine C. Marshall and Peggy M. Irish. Guided tours and on-line presentations: How authors make existing hypertext intelligible for readers. In *ACM Hypertext'89 Proceedings*, pages 15–26, 1989.
- [MYBM91] Allan MacLean, Richard M. Young, Victoria M. E. Bellotti, and Thomas P. Moran. Questions, options, and criteria: Elements of design space analysis. *Human-Computer Interaction*, 6(3,4):201–250, 1991.
- [NCK⁺92] Christine M. Neuwirth, Ravinder Chandhok, David S. Kaufer, Paul Erion, James Morris, and Dale Miller. Flexible diff-ing in a collaborative writing system. In *Proceedings of ACM CSCW'92 Conference on Computer-Supported Cooperative Work*, pages 147–154, 1992.
- [NG84] Joseph D. Novak and D. Bob Gowin. *Learning how to learn*. Cambridge University Press, 1984.
- [NKCM90] C.M. Neuwirth, D.S. Kaufer, R. Chandhok, and J. Morris. Issues in the design of computer support for co-authoring and commenting. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)*, Los Angeles, California, 1990. ACM Press.
- [Nov90] J.D. Novak. Concept mapping: a useful tool for science education. *Journal of Research in Science Teaching*, 27(10):937–49, dec 1990.

- [Pea93] Roy Pea. The collaborative visualization project. *Communications of the ACM*, 36(5):60–63, May 1993.
- [Pia32] J. Piaget. *The moral judgement of the child*. Routledge & Kegan Paul, London, 1932.
- [Pia77] J. Piaget. *The equilibration of cognitive structures*. University of Chicago Press, 1977.
- [Qui67] Ross M. Quilian. Word concepts: a theory and simulation fo some basic sematic capabilities. *Behavioral Science*, 12:410–30, 1967.
- [RR92] Wolff-Michael Roth and Anita Roychoudhury. The social construction of scientific concepts or the concept map as conscription device and tool for social thinking in high school science. *Science Education*, 76(5):531–557, 1992.
- [Rum80] D.E. Rumelhart. Schema: the building block of cognition. In R. Spiro, B.C. Bruce, and W.F. Brewer, editors, *Theoretical issues in reading comprehension*. Lawrence Erlbaum Associates, 1980.
- [SA77] Roger Schank and Robert Abelson. *Scripts, plans, and goals, and understanding: an inquiry into human knowledge structure*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.
- [SB93] Marlene Scardamalia and Carl Bereiter. Technologies for knowledge-building discourse. *Communications of the ACM*, 36(5):37–41, May 1993.
- [Sch75] R. C. Schank. Conceptual dependency theory. In R. C. Schank, editor, *Conceptual Information Processing*, pages 22–82. North-Holland and Elsevier, Amsterdam and New York, 1975.
- [Sel93] Jack Selzer, editor. *Understanding scientific prose*. University of Wisconsin Press, 1993.
- [SFG⁺87] Neil Stillings, Mark Feinstein, Jay Garfield, E.L. Rissland, D.A. Rosenbaum, S.E. Weisler, and L.Baker-Ward. *Cognitive science: an introduction*. MIT Press, 1987.
- [Sim90] Herbert Simons, editor. *The rhetorical turn: invention and persuasion in the conduct of inquiry*. University of Chicago Press, 1990.
- [Swa90] Kishore Swaminathan. *RA: a memory organization to model the evolution of scientific knowledge*. PhD thesis, University of Massachusetts at Amherst, 1990. also COINS Technical Report 90-80.

- [SWF87] John B. Smith, Stephen F. Weiss, and Gordon J. Ferguson. A hypertext writing environment and its cognitive basis. In *ACM Hypertext' 87 Proceedings*, pages 195–214, 1987.
- [TI87] Randall H. Trigg and Peggy M. Irish. Hypertext habitats: Experiences of writers in Notecards. In *ACM Hypertext' 87 Proceedings*, pages 89–108, 1987.
- [Tic92] Walter Tichy. Programming-in-the-large: past, present, and future. *1992 IEEE 14th International Conference on Software Engineering*, pages 362–7, 1992.
- [Tja93] Danu Tjahjono. CSRS: a new approach to software review. Technical Report CSDL-TR-93-04, University of Hawaii, Department of Information and Computer Sciences, 1993.
- [Tou58] S. Toulmin. *The use of argument*. Cambridge University Press, 1958.
- [Tri88] Randall H. Trigg. Guided tours and tabletops: Tools for communicating in a hypertext environment. In *Proceedings of 1988 Conference on Computer-Supported Cooperative Work*, pages 216–226, 1988.
- [TRJ84] Steven Toulmin, Richard Rieke, and Allan Janik. *An introduction to reasoning*. Macmillan, New York, 2nd edition, 1984.
- [vDK83] T.A. van Dijk and W. Kintsch. *Strategies of discourse processing*. Academic Press, 1983.
- [VTY83] J.F. Voss, S.W. Tyler, and L.A. Yengo. Individual differences in the solving of social science problems. In E.E. Dillon and R.R. Schmeck, editors, *Individual differences in cognition*, pages 205–32. Academic Press, 1983.
- [Wan93a] Dadong Wan. CLARE 1.4.7 design document. Technical Report CSDL-TR-93-24, University of Hawaii, Department of Information and Computer Sciences, September 1993.
- [Wan93b] Dadong Wan. CLARE User's Guide. Technical Report CSDL-TR-93-15, University of Hawaii, Department of Information and Computer Sciences, September 1993.
- [WJ92] Dadong Wan and Philip M. Johnson. Supporting scientific learning and research review using COREVIEW. In *Proceedings of the 1992 AAAI Workshop on Communicating Scientific and Technical Knowledge*, July 1992.

- [Woo85] W. A. Woods. What's in a link: Foundations for semantic networks. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, pages 217–241. Kaufmann, Los Altos, CA, 1985.
- [YHMD88] Nicole Yankelovich, Bernard J. Haan, Norman K. Y. Meyrowitz, and Steven M. Drucker. Intermedia: the concept and the construction of a seamless information environment. *Computer*, January 1988.