MODELS OF PROFESSIONAL WRITING PRACTICES WITHIN THE FIELD OF COMPUTER SCIENCE

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

by

THOMAS KEVIN ORR

DISSERTATION ADVISOR: DR. W. WEBSTER NEWBOLD

BALL STATE UNIVERSITY

MUNCIE, INDIANA

DECEMBER 1995

UMI Number: 9611747

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

MODELS OF PROFESSIONAL WRITING PRACTICES

WITHIN THE FIELD OF COMPUTER SCIENCE

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

by

Thomas Kevin Orr

Approved by:

Committee Chairman	11/15/95 Date
Committee Member	11/15/95 Date
Committee Member	11/15/95 Date
Committee Member	11/15/95 Date

Ball State University

Muncie, Indiana

December 1995

ABSTRACT

Twenty-five computer scientists representing thirteen language groups at a highly respected university in Japan were surveyed and interviewed to identify the English writing products and processes characteristic of their professional work. Twenty-two major genres were identified and grouped according to purpose. Fourteen significant contextual factors that affected writing production and, ultimately, success in the field of computer science were also identified. Several models were proposed to provide an overview of the writing practices and products potentially illustrative of writing in the computer science field.

In the second research phase, two detailed case studies were conducted involving one native and one non-native speaker of English to identify how English language proficiency affected professional practice. *Efficiency* was a major concern for both subjects but employed different strategies to make their work as efficient and productive as possible.

The native speaker, for example, employed skillful use of the computer to gather, create, and store data that could be efficiently "chunked" and assembled into papers, either for publication or for obtaining valuable feedback from other professionals. The subject also thought a great deal about the expectations of his readers as well as the referees and editors who judged the quality of his work. He also revised his work extensively and recruited other professionals locally and abroad to assist him in refining his texts.

The non-native speaker employed many of the same computer strategies for gathering and managing information; however, the added difficulty of functioning professionally in English severely limited his work pace and left too little time before deadlines to revise or solicit the amount of feedback he desired from his peers. His most effective strategy was co-authoring papers with native speakers of English who could handle most of the final editing and revising.

In the final chapter, results of the surveys, interviews, and case studies were illustrated graphically in an algorithmic flowchart of professional writing practice, and educational applications for writing instruction as well as recommendations for additional research were also suggested.

Contents

Chapter 1

Introduction

1.1 The Primacy of English

Ever since the geniture of the first modern computers in the 30s and 40s,[1] the ubiquitous spread of computers into every field of study and every area of human activity has begun to transform the entire planet, making information one of the chief commodities. Toffler (1970, 1980, 1983) calls this new era the *Information Age* with *information* being the newest addition to *land*, *labor*, and *capital*, the three traditional economic cornerstones (Capron, 1992). Computers are altering the way people learn, the way people work, and the way people form communities. Particularly, since the evolution of Internet, people have begun to group themselves more and more by special interests rather than by geographic locality. In fact, it is now quite common for individuals living in neighboring apartments or employed at the same locale to remain virtual strangers while at the same time interacting meaningfully and productively

[1]Though various computing devices appeared prior to the 20th century, ranging from the abacus of ancient Babylon (Goetz, 1988) to the Analytical Engine developed by Charles Babbage in England during the 1820s (Capron, 1992), the modern age of computers can be said to begin with the vacuum tube Atanasoff-Berry Computer invented in 1939, the Z3 programmable computer developed by Konrad Zuse in 1941, IBM's sequence-controlled calculator called Mark I in 1944, and the first completely electronic computer labeled ENIAC (Electronic Numerical Integrator and Computer) designed by John von Neumann in 1946 (Kinkoph et al, 1994; Brookshear, 1994).

with like-minded peers on the other side of the planet.

One obvious effect from this revolutionary restructuring of human communities and human knowledge is the privileging of English as the international language of communication. With assistance from computers, people can interact and exchange vast quantities of information cheaply and efficiently in a matter of seconds or in real time itself. This remarkable capability has, most significantly, internationalized academic disciplines and intensified the creation and exchange of knowledge on a global scale hitherto unforeseen. The evolution of English as the most widely used language for this knowledge exchange is primarily due to the high concentration of excellent research facilities and the rapid advancement of computer technology in the United States, two factors that have continued to make the U.S. very attractive to international graduate students and scholars interested in furthering their education and expanding their research opportunities. Since many of the world's scholars are educated in English and much of the world's research is carried out in English, it is natural that English has become the preferred language for educated exchange around the world; it is the one language that most scholars have in common. As a result, English has become one of the most widely used languages in the academic world for disseminating and archiving its knowledge.

1.2 The Development of ESP

Most of the world's researchers, however, do not speak English natively and value any assistance they can receive that will improve their ability to function in English more professionally. This has engendered a boom in English language education and created a new breed of language-teaching specializations known most broadly by the cover term English for Specific Purposes (ESP) and more specifically by its many subdomains such as English for Science and Technology (EST), English for Medical Purposes (EMP), English for Business Purposes (EBP), English for Academic Purposes (EAP) and a rapidly growing host of others.

Generally, the learners served by this English language instruction are non-native speakers (NNSs), both seasoned professionals and university students just entering their fields of study. There are parallel movements in education, however, for native speakers (NSs). The Writing Across the Curriculum (WAC) program, in the United States, and studies of science and business within the fields of sociology, history, and psychology are prime examples. The most active area of research for specific instructional applications, however, that is designed to prepare NNSs to use English successfully within their disciplines, is ESP.

1.3 ESP: Definitions and Goals

ESP has continually been defined by its practitioners since its beginnings in 1960s. The following descriptions represent two of the most well-known.

1.3.1 Hutchinson and Waters

Tom Hutchinson and Alan Waters (1987), in a comprehensive description of ESP in their book *English for specific purposes: A learner-centered approach*, end their third chapter with a description of what ESP is *not*, quoted here in full (p. 18).

a) ESP is *not* a matter of teaching 'specialized varieties' of English. The fact that language is used for specific purposes does *not* imply that it is a special form of the language, different in kind from other forms. Certainly, there are some features which can be identified as 'typical' of a particular context of use and which, therefore, the learner is more likely to meet in the target situation. But these differences should not be allowed to obscure the far larger area of common ground that underlies all English use, and indeed, all language use.

b) ESP is *not* just a matter of Science words and grammar for Scientists, Hotel words and grammar for Hotel staff and so on. When we look at a tree, we see the leaves and branches, but there is much more to the tree than just these–much of it hidden from view inside and beneath the tree. The leaves do not just hang in the air: they are supported by a complex underlying structure. In the same way there is much more to communication than just the surface features that we read and hear. We need to distinguish, as Chomsky did with regard to grammar, between *performance* and *competence*, that is between what people actually do with the language and the range of knowledge and abilities which enables them to do it.

c) ESP is *not* different in kind from any other form of language teaching, in that it should be based in the first instance on principles of effective and efficient learning. Though the content of learning may vary there is no reason to suppose that the process of learning should be any different for the ESP learner than for the General English learner. There is, in other words, no such thing as an ESP methodology, merely methodologies that have been applied in ESP classrooms, but could just as well have been used in the learning of any kind of English.

They conclude by saying that "ESP must be seen as an *approach* not as a *product*....Understood properly, it is an approach to language learning, which is based on learner need (p. 19)." It is not concerned with a particular variety of language, but with how the learners must learn to use particular aspects of language in specific contexts.

1.3.2 Strevens

Peter Strevens (1988), another active spokesman for ESP, offers the following definition of ESP.

> A definition of ESP needs to distinguish between four absolute and two variable characteristics:
>
> 1) Absolute Characteristics
>
> ESP consists of English language teaching which is
> a) designed to meet specified needs of the learner;
> b) related in content (i.e., in its themes and topics) to particular disciplines, occupations, and activities;
> c) centered on the language appropriate to those activities in syntax, lexis, discourse, semantics, etc., and analysis of this discourse; and
> d) in contrast with "General English."
>
> 2) Variable Characteristics
>
> ESP may be, but is not necessarily
> a) restricted as to the language skills to be learned (e.g., reading only) and
> b) not taught according to any pre-ordained methodology.
>
> The claims for ESP are
> a) being focused on the learner's need—wastes no time;
> b) is relevant to the learner;

c) is successful in imparting learning; and

d) is more cost-effective than "General English." (pp. 1-2)

1.3.3 Summary

Clearly, ESP is goal-directed language education designed to prepare language learners to use English successfully in specific contexts to achieve specific purposes. As such, the primary concerns of its practitioners are with the analysis of student needs and the development of appropriate curricular applications (Johns & Dudley-Evans, 1991). These concerns, of course, must be worked out differently in each educational context. The English language needs of a 25-year-old employee of a Swiss pharmaceutical company preparing for work at her company's London office will differ considerably from an 18-year-old Indonesian boy enrolled in agriculture at the University of Nebraska. Both are preparing to enter vastly different discourse communities for the accomplishment of different purposes. It is the role of the ESP researcher-instructor to discover what tasks the language learner must prepare to accomplish in his or her target English culture and to prescribe the appropriate instructional activities that will enable success. How to go about these seemingly simple tasks has been the subject of much debate and study for the past 30 some years with many questions left yet unanswered.

1.4 This Study

The topic of this particular dissertation concerns two of the most important issues central to the field of ESP: needs analysis and curricular applications. The focus, however, will primarily be on the first. The educational context for the study falls within a narrower domain of ESP called EST (English for Science and Technology), more specifically, ECS (English for Computer Science). The research proposal and methodology for this study are outlined in Chapter 2, research results are detailed in Chapters 3 through 6, a discussion of the results and curricular applications are presented in Chapter 7, and References and Appendices follow at the end.

Chapter 2

The Research Context

2.1 English for Science and Technology

One of the earliest branches of ESP was English for Science and Technology (EST), a cover term for all research and instructional activity designed to understand and support the effective use of English in scientific and technological fields. The term EST was apparently first coined by Selinker in the mid-sixties during his tenure at the University of Washington and originally meant "the written discourse of scientific and technical English" (Trimble, 1985 p. 2) where Selinker and Trimble jointly developed a reading/writing course for nonnative English speakers pursuing undergraduate and graduate degrees in "engineering, the physical and natural sciences, pre-medicine and dentistry, nursing, nutrition, and home economics" (Trimble, 1985, p. 137). As other programs developed, however, and as more educators/researchers joined the work, research efforts expanded beyond Selinker and Trimble's analysis of scientific English in college textbooks and popular science magazines (i.e., *Scientific American*) to include inquiry into scientific citations (e.g., Bavelas, 1978), reports (e.g., Bazerman, 1984), slide usage (i.e., Dubois, 1980), student writing assignments (e.g., Horowitz, 1986), scientific verb tense (e.g., Malcolm, 1987), course descriptions (e.g., Lenze, 1988), and other diversified usage of scientific English in a wider range of scientific and technology-related disciplines. Research efforts also broadened to encompass more language-learning theory (e.g., Alexander & Judy, 1989; Wittrock, 1985) and language-learning applications (e.g., Holes, 1984; McKenna, 1987) and began to draw upon a wider circle of information from other fields. Consequently, the acronym EST has now evolved to identify far broader concerns. It can be understood to include all research and pedagogical activities related to English language learning and usage in scientific and technical fields. The most frequent appearance of the term, however, continues to surface in English as a Second Language (ESL) or English as a Foreign Language (EFL) communities where the educational concern is primarily that of nonnative speakers.

2.2 Composition Research in EST

The focus of research and instruction in EST from the very beginning has primarily been in the area of writing. This is understandable since the production of text is central to the work in scientific and technical fields. Much of the early work was of a textual-linguistic nature, but recent work has broadened to include socio-rhetorical and psychological aspects. Much of the work to date, however, has been piecemeal, with too few studies attempting to build a comprehensive model of writing practices as they are *situated* (Brown, Collins, & Duguid, 1989; Rogoff & Lave, 1984) in fields of science or technology that can shed light on the context of writing decisions and preferences within a profession. All too often, the practice has been to identify features of a text and then ask students to imitate the features in their own writing. Though studies of textual products have value in identifying various language features conventional to a particular discourse community, most have not gone far enough to assess why particular features are preferred according to what contextual constraints.

The following is a sample of the kinds of features in scientific texts that were studied during the 70s and 80s when textual analysis was at its zenith.

- authorial comment (Adams Smith, 1984)
- citation patterns (Bavelas, 1978; Dubois, 1988)
- compound nominals (Salager, 1984)
- function of grammatical alternation (Pettinari, 1982)
- hedging (Rounds, 1982)
- information structure (Bruce, 1983)
- introductions (Cooper, 1985)
- lexis (Inman, 1978; Bramki & Williams, 1984))
- modal usage (Lackstrom, 1978; Ewer, 1979)
- noun phrases (Dubois, 1982)
- paragraph development (Lackstrom et al, 1973; Weissberg, 1984)
- personal pronouns (Ard, 1983)
- results statements (Swales & Najjar, 1987)
- role of definitions (Darian, 1982)
- schema (Bazerman, 1985)
- tense (Oster, 1981; Een, 1982; Heslot, 1982; Malcolm, 1987)
- *that*-nominals (West, 1980)
- topic sentences (Popken, 1987)
- verb forms (Wingard, 1981)

Not all researchers in EST, however, have been concerned with textual features of such narrow proportion. Others have looked at texts as a whole and attempted to describe and categorize them according to broader considerations. Three of the most well-known are the Trimble/Selinker team and John Swales.

Trimble (1985) describes the work that he and Selinker did at the University of Washington in the 1970s and 80s when they studied the English of science textbooks and articles in *Scientific American.* The result of their effort was the creation of an **EST Rhetorical Process Chart** with four descriptive levels of rhetorical activity.

Level A, in their chart, described the objectives of the total discourse:

1) detailing an experiment,

2) making a recommendation,

3) presenting new hypotheses or theory, and

4) presenting other types of EST information.

This was followed by Level B which described the general rhetorical functions that developed these objectives, such as

1) stating the purpose,

2) reporting past research,

3) stating the problem,

4) presenting information on apparatus used in an experiment, and

5) presenting information on experimental procedures.

Level C continued with a description of specific rhetorical functions that developed the functions of Level B. These were

1) descriptions,

2) definitions,

3) classifications,

4) instructions, and

5) visual-verbal relationships.

Level D, then, divided these rhetorical functions further by showing the relationship within or between the functions listed in Level C. These relationships consisted of various ways of ordering information, such as

1) time,

2) space,

3) causality and result,

or patterning information, such as

1) causality and result,

2) order of importance,

3) comparison and contrast,

4) analogy,

5) exemplification,

6) illustration.

What is interesting in this early writing research in EST is the fact that this analysis and description were made independent of knowledge about the context. There was, apparently, no contact with the writers of the texts under study to learn *who* they were writing for and *how* they put these texts together. There was no investigation of context to demonstrate *why* specific "rhetorical" features where selected to accomplish *what* particular objectives. In fact, one might even seriously question whether college textbooks and articles in *Scientific American* are representative enough of the writing a scientist does to have much value at all in orienting university students to the written documents and practices common to their chosen discipline.

A more recent researcher, of much renown, whose work has prompted a flood of research after he first introduced his method of textual analysis, is John Swales. Swales' method of looking at writing (and reading) instruction for EST involves the classification of texts into professional genres which, he explains, are

> class[es] of communicative events, the members of which share some set of communicative purposes. These purposes are recognized by the expert members of the parent discourse community, and thereby constitute the rationale for the genre. This rationale shapes the schematic structure of the discourse and influences and constrains choice of content and style. Communicative purpose is both a privileged criterion and one that operates to keep the scope of a genre as here conceived narrowly focused on comparable rhetorical action. In addition to purpose, exemplars of a genre exhibit various patterns of similarity in terms of structure, style, content, and intended audience. If all high probability expectations are realized, the exemplar will be viewed as prototypical by the parent discourse community. The genre names inherited and produced by discourse communities and imported by others constitute variable ethnographic communication, but typically need further validation. (1990, p. 58).

Here again, the focus is on the study of scientific texts, though clearly those more commonly produced by scientists. He gives particular attention to research articles (RAs) and the rhetorical moves or activities that scientists take within such texts as demonstrated by studies of the documents themselves. Swales work *is* impressive and has made a successful transition from knowledge for EST professionals to knowledge for NNSs who are learning to write at the graduate school level in American universities.[1] However, one thing still lacking in his work, no matter how impressive,

[1]See Swales, J. & Feak, C. (1994). ***Academic writing for graduate students: A course for nonnative***

is a comprehensive view of genres within their particular professions. EST professionals still do not know enough about how these documents develop within different professional environments. Too much work in EST has been carried out apart from interaction with the authors who construct these texts. Too much work has detailed what researchers assume is *context-independent* writing without enough input from "insiders" to know how much of their decision-making is *context-dependent.* Interestingly enough, Swales (1985) and others (Ramani et al, 1988; Johns & Dudley-Evans, 1991; Bhatia, 1993) have hinted at this weakness in EST writing studies and call for better collaboration between experienced "insiders" within a scientific discipline and informed "outsiders" who know what language features to look for.

What is needed now in EST is research beyond the textual level. What is needed is research of a broader nature that can fully encompass all of a scientist's work, experience, and professional goals that can better contextualize document production within professional fields of endeavor. What is needed is a model of writing practices that, as one leading practitioner suggests, fully incorporates

> (1) the L2 writer (the person–in terms of personal knowledge, attitudes, and characteristics; cultural orientation, language proficiency; motivation, etc.–as well as the process); (2) the L1 reader–perhaps the primary audience for academically oriented, college-level ESL writers (with regard to the person and the reading process); (3) the L2 text (in terms of genre, aims, modes, discourse structures, intersentential phenomena, syntax, lexis, and print-code features); (4) the contexts for L2 writing (cultural, political, social, economic, situational, physical); and (5) the interaction of these elements in a variety of authentic ESL settings. (Silva, 1990).

speakers of English. Ann Arbor, MI: University of Michigan Press.

What is needed is a comprehensive model of the unique writing culture in each field of science or technology that can adequately account for common practices within the profession and better inform English writing instruction for NNSs who wish to join the profession. To date, no studies have fully accomplished this.

2.3 Writing Models Outside EST

Outside EST, the only research that has sought to build models of writing has been that conducted by composition researchers with a strong cognitive science orientation. Two of the most successful research teams in this category have been the Flower and Hayes team and the Bereiter and Scardamalia team. It is appropriate that we briefly look at their work.

2.3.1 Flower and Hayes

Linda Flower, a compositionist, and John Hayes, a cognitive psychologist, joined forces at Carnegie-Mellon to study the composing processes of writers via protocol analysis in order to better understand what writers do and to discover how writers could learn to do it better. One of the applications of their research was to produce a university writing course with the guiding objective to "make unconscious actions a little more conscious: to give writers a greater awareness of their own intellectual processes, and therefore the power and possibility of conscious choice" (Flower, 1985, p.vii). The model they constructed divided a writer's world into three realms: a *task environment*, where writing assignments were received and produced; the *writer's long term memory*, which the writer accessed for knowledge of the topic, the audience, and writing plans; and the *writing process*, where the writer planned, translated, and reviewed his or her work (Hayes & Flower, 1980). This model, later, became the foundation for a writing textbook (Flower, 1985) that offered students nine steps of multiple strategies to aid them in their writing process. These included

1. exploring the rhetorical problem
2. making a plan
3. generating ideas
4. organizing ideas
5. analyzing the needs of the readers
6. transforming *writer-based* prose to *reader-based* prose
7. reviewing the paper and its purpose
8. testing and editing the paper
9. editing for connections and coherence

This description of the writing process and its translation into a recipe for good writing was quite popular in freshman composition courses in the United States where assignments are of a fairly generic nature with little resemblance to the writing that actually takes place in the sciences and technical fields. Unfortunately, neither the model nor its instructional application is reflective of what happens in scientific disciplines nor applicable to the specific needs of NNSs who wish to approximate the professional practices of their field's most established membership.

2.3.2 Bereiter and Scardamalia

Another team of researchers that have studied the writing process and posed models to account for the system are Carl Bereiter and Marlene Scardamalia of the Ontario Institute for Studies in Education. In their text (1987), Bereiter and Scardamalia construct two models of composing processes which they claim writers customarily employ.

The first model is what they term a *knowledge-telling model.* Here a writer will take a mental representation of a writing assignment, locate topic and genre identifiers, construct memory probes, retrieve content from memory using the probes, run tests of appropriateness, write notes or a draft, and update the mental representation of the text, all the while drawing continually upon content knowledge and discourse knowledge to inform decisions. This model, they claim, portrays the writing process most frequently used by young, inexperienced writers who view writing as simply emptying out content on paper. Here, writing is photographic; it records an image of what already exists in the mind in symbolic form.

Their second model is what they term a *knowledge-transforming model.* Writers who employ this process construct knowledge as they continually recycle through a complicated system of analysis, goal setting, and translation between two problem spaces (a *content* problem space and a *rhetorical* problem space) where knowledge is built and interchanged until personal goals are satisfied and a piece of writing emerges. The product of this writing process is content that never existed in the mind at the onset. It evolved from the elements that already existed and/or were triggered by the thinking-writing process into knowledge that the author most likely didn't anticipate.

Unfortunately, these two models account for only a portion of writing that *may* actually take place in a professional context. It is what Shriver (1992) would call *context-independent* writing and based on the assumption that writers in all disciplines write in the same way and employ the same method for each writing task they encounter. They do not account for all factors involved in a professional writer's decision-making.

2.4 Problem with Existing Models

Evident from the work described above, these models are not adequate for detailing all that goes on within a particular discourse community. They are too narrow to encompass an entire writing culture and not detailed enough to demonstrate how the writing practices in one discipline will differ from another. What is needed is a much broader picture of professional writing set in the fullest context of elements relative to the process. What is needed is a model that fully includes the writing contexts specific to each profession and how members of that profession commonly write under these conditions. This dissertation will attempt to satisfy this need.

2.5 Research Proposal

The goal of the research documented in this dissertation is the construction of a model that closely approximates the actual professional writing practices within a scientific profession, specifically the field of computer science. The model must include all significant matters relevant to the production of the text within computer science, must be broad enough to portray the common practices that unite the international computer science community and must account for differences between individuals and native-language groups. When complete, the model should provide a complete account of the rich professional context within which computer science writing occurs.

The value of such a project is clear and is described in the three sections that follow.

2.5.1 Value to Writing Researchers

This study serves writing researchers in ESP,[2] EST,[3] EAP,[4] or WAC,[5]

1. by showing how a writing discourse community can be studied and its practices illustrated to incorporate a fuller context of the writing culture under study and
2. by providing insights into the specific writing practices of a particular writing community that can
   a. serve for comparisons with other writing communities and
   b. serve as a heuristic for research from other angles in the field.

2.5.2 Value to Writers in Computer Science

This study serves writers in computer science (both professionals and students)

1. by showing what writing patterns and flows of activity are conventional in the field of computer science and
2. by suggesting which routes are the most efficient to follow.

2.5.3 Value to Writing Educators

And finally, this serves writing instructors who offer writing support to writers in computer science (and perhaps those in other fields as well) by positing a flow chart of professional writing-related activities that

[2]English for Specific Purposes
[3]English for Science and Technology
[4]English for Academic Purposes
[5]Writing Across the Curriculum

1. can be used diagnostically by comparing personal writing practices with those of professionals to see where processes differ and

2. can be used instructionally by providing a template of professional activities that teachers can assist students/clients to fill in with the appropriate knowledge and experience to expand the students'/clients' repertoire of options they can negotiate skillfully.

2.6 Research Methodology

2.6.1 Method

The research in this dissertation employs *Formalist Inquiry*, a research method described by North in his book *The making of knowledge in composition: Portrait of an emerging field* (1987). Here under North's list of different kinds of *Researchers* (*experimentalists, clinicians, formalists, and ethnographers*), it is the *formalists* who seek to develop models or paradigms to account for multidimensional human activities (which Nunnally, 1978, by the way, numbers around 400). The value of these, of course, is that they enable us to get a handle on rather abstract human processes by positing possible elements in a process and how these elements relate to one another. These can also be used heuristically by suggesting where empirical studies might begin to test the accuracy of the models to see how adequately they portray the empirical world. Formalist models can never *explain* or *describe* how writers write within contexts—nothing can. However, they *can* be used to build theories of how writing occurs in specific contexts that people can build assumptions on in order to function intelligently. These can continually be tested against real-world experience and refined until they appear to approximate reality. Without such assumptions, neither writing nor writing instruction could exist in any meaningful, task-directed form; it would simply exist in chaos with no expected outcomes.

The means of data collection for this inquiry, that informs and refines the construction of this writing paradigm, is *multimodalic* (Lauer & Asher, 1988), for it incorporates both surveys and interviews in both ethnographic and case study modes as well as various aspects of textual analysis appropriate to this project. Chapter 3 documents the process of identifying writing events and related contextual elements in the professional computer science discourse community; Chapter 4 proposes a potential model of computer science writing and its context; Chapter 5 presents the case study of one native English writer functioning within the model; Chapter 6, presents the case study of one nonnative English writer functioning within the model; and then Chapter 7 concludes this study by refining the model and discussing the implications for instructional activity and for future research.

2.6.2 Site of Application

The site chosen for this research is the University of Aizu, a new university in Japan offering degrees solely in computer science hardware and software engineering. The location is ideal for a study of professional writing practices in computer science; for it employs a broad range of professionals from approximately 16 nations and 13 language groups, the faculty consist of both seasoned professionals and recent recruits, and there is an equal balance of experience among the faculty in academic and corporate writing experience. In addition, since the university was established in 1993, nearly all faculty arrived on campus to begin work at the same time under many of the same conditions. The University of Aizu is also a research university and places high value on writing and publication, activities that the computer science faculty have actively engaged in from their first year on campus.[6] The university is also set firmly in an EST environment where students—many of which will become future computer scientists—are all native speakers of Japanese.

One final advantage of selecting the University of Aizu for this study is that the researcher conducting this study is employed at this institution in the Center for Language Research, a facility for research and instructional support for both students and professionals in computer science, and is conveniently a computer science *outsider* with daily access to the computer science discourse community and its *insiders*, both on campus and off. Consequently, the research location is advantageous as both a representative sample of the international academic computer science community and as a research community easily accessible to the researcher.

[6]See the *1993 Annual Review: School of Computer Science and Engineering*, issued by the University of Aizu, for abstracts of the writing and research activities of the hardware and software faculty during their first year of employment.

Chapter 3

Identifying the Relevant Elements

3.1 Fundamental Questions

If we are to begin to understand the normal writing activities common to computer scientists,[1] this research must attempt to answer the following fundamental questions:

To Guide the Search for Data

- What writing activities can be found among the professional activities of the representative computer scientists in this study?
- What are the intended purposes of these writing events?
- What important factors shape the writing process and the written products that result?

To Guide the Concern for Application

[1]In this study, I define *computer scientists* as those individuals within the computer field who hold masters and/or doctorates in computer science or a related field, devote the majority of their time to research in computer software and/or hardware, and regularly attempt to disseminate their research findings to other computer scientists through professional computer science mediums such as conferences and scholarly publications (e.g., IEEE, ACM, etc.).

- How can the writing and related contextual factors be illustrated in graphic form (i.e. a model)?
- How can this model be used to aid further research and to aid writing instruction for university students who are training to become successful computer scientists?

Chapters 3 through 6 will document the research carried out in pursuit of answers to questions one through four, and Chapter 7 will address question five.

3.2 Research Method for Phase One

The initial information-gathering phase carried out during the first two and one-half years at the University of Aizu incorporated a series of six major questionnaires and four minor questionnaires distributed to faculty members in the Department of Computer Science Software and the Department of Computer Science Hardware. The questionnaires were distributed and responses were collected primarily through the university's electronic mail network. Faculty members who failed to respond to the initial e-mailings addressed to the alias *all-software* and *all-hardware* were often sent the questionnaires a second time with personalized solicitations for response. Input was also gathered informally through individual e-mail correspondence, numerous private interviews on campus throughout the first two and one-half academic school years and during the three-day/two-night freshman student orientation camp at the beginning of the 1994 school year in April. This data was then filed electronically according to the survey number and printed out for compilation in notebooks organized according to contributors. Data was then entered in a spreadsheet to better study both individual responses and overall patterns.

Because the university was new, however, and because of the tremendous amount of time and energy required by the faculty to design course instruction, organize and set up the research laboratories, establish policies and mechanisms for efficient university business, continue their research and publishing activities, in addition to taking care of numerous details to get their families settled in a new locale and their children adjusted to schooling in a foreign country and in a foreign language, the feedback was naturally sporadic and uneven. Some faculty members contributed data enthusiastically to the project immediately at every request, others responded when time permitted, and several gave no response at all. It was observed, however, that those who did respond to requests appeared to value writing a great deal, had much to say about it, and proved to be both active and successful in it. It was the input from these individuals that provided the most insight into writing activities characteristic of the computer science profession.

Results from the questionnaires and interviews are presented in the sections that follow. Since the researcher sought to gather information for a range of research purposes far broader than this present dissertation, only information relevant to this study has been recorded in the following pages. Specific survey questions that were used for this particular dissertation are listed in Appendix A.[2]

[2]In order to respect the privacy of both those who contributed data and those who did not, I have chosen not to identify responses with specific names of faculty members nor with specific nationalities unless I have been given permission to do so, as with the two case studies that follow in Chapters 5 and 6. In a research university where research and publication are two of the most important measures of success and value to the institution, it seems wise that professionals who struggle with writing and publishing in English for the international computer science community (particularly NNWs) should not be singled out because of their current struggles, but rather assisted by the input provided here by their peers. Privacy is of particular importance here at the University of Aizu, since all faculty will be evaluated for tenure at the end of their third year of employment, which will take place for the majority of faculty shortly after this dissertation is made public. Consequently, utmost care has been taken to allow respondents to provide accurate information and honest opinions without fear of political consequences.

3.3 Identifying Professional Writing

The first and most important information to be elicited concerned the specific writing activities that the local sample of computer scientists regularly engage in to accomplish their work. In response to both written and oral questioning, respondents said that they normally produced the following texts. Specific terms used by the computer science faculty have been retained and are listed here randomly at this phase of investigation.

- e-mail
- technical reports
- journal articles
- miscellaneous business letters
- proposals
- articles for professional newsletters or SIG[3] publications
- conference papers/proceedings
- research reports
- descriptions of laboratories or departments
- announcements
- conference/seminar posters
- abstracts
- minutes of meetings
- research notes
- working papers (preliminary drafts of technical reports, etc.)
- biographical sketches
- OHP transparencies
- textbooks
- software documentation

[3] Special Interest Group

- instruction manuals
- chapters in books
- reviews
- examinations
- course handouts
- PR (public relations) brochures
- progress reports
- university or personal World Wide Web pages

Though more detailed information about each item was gathered both in the initial surveys and in follow-up interviews, it is appropriate that these be considered later.

3.4 Identifying Other Relevant Elements

In addition to the specific varieties of writing listed above, the respondents also commented that the following related matters were important parts of the writing context within which they normally wrote.

3.4.1 Research

The first and most obvious element relevant to writing that emerged in response to the researcher's questions was the topic of *research.* According to all who responded, writing is a major subset of the all-encompassing research effort, the focus of each scientist's work. Without research, there would be very little worth writing about; and without written dissemination, there would be no point in doing research. Knowledge created and never disseminated and/or applied is meaningless—except, perhaps, in building one's self-esteem which others, unfortunately, can neither validate nor appreciate. As one respondent put it, "If I don't do research, I have nothing to write about." Clearly, in developing a model of writing in the professional computer science community, research is a significant feature of the computer science context.

3.4.2 Reading

Another professional activity that appeared to be an important and necessary part of every computer scientist's work was the activity of reading. Though respondents claimed that they needed to read a number of different items to carry out their work (i.e., instructions, university announcements, the minutes of meetings, etc.), the most important reading material consisted of professional journals, newsletters, books, and conference proceedings in order to keep abreast with what is new or what has already been done in their field. In fact, faculty members claimed they read anywhere from 1,030 to 25,500 pages of professional literature each year, with the average being 5,558 pages calculated from a pool of 25 respondents.

As expected, English was the most commonly read language even though the majority of computer scientists who responded were not native English speakers. As one Asian professor put it, "anything worth knowing about in computer science is published in English." As a matter of fact, no respondent reported reading more pages in another language than he or she read in English.

Of course, native English speakers had the clear advantage when it came to reading English. They read on average 10,667 pages of professional literature in English each year in comparison with native speakers of 12 other language groups who reported reading 3,308 pages of English each year. Only 1 of the 25 computer scientists read only English. Most of the computer faculty read professional computer science literature in other languages too–anywhere from 2 to 7 other languages, the most common number of languages read being 3.

Clearly, reading professional computer science literature, particularly that which is published in English, appears to be a very important and time-consuming activity that many professionals regularly engage in. Based upon follow-up questions sent to or asked of many of those who reported their reading statistics, reading was the number one channel for professional input, with conferences, seminars, and dialogues with other professionals playing far less important roles in their accumulation of professional knowledge.

It was also mentioned several times, and will be confirmed later in the case studies, that this continual exposure to professional literature was the most influential source of knowledge about professional writing in computer science—the continual exposure to examples.

3.4.3 Previous Writing Instruction

When faculty members were asked how they learned to write the kinds of English texts they normally wrote, they gave the following responses:[4]

- from university writing course(s)
- from professors or senior research associates who mentored them
- from reading many of the same kinds of documents and imitating them
- from consulting various writing handbooks
- from comments provided by reviewers, editors and/or colleagues

[4]The reader should be reminded that the purpose of this present research is to identify significant features of the professional computer science writing context, not to determine which educational practices are *most productive*, though future research efforts at the Center for Language Research will likely include these goals.

Though most of those who responded learned to write through a combination of all of the experiences listed above, many claimed the large volume of papers that they read made exposure to examples the most influential source of instruction for them. Several faculty commented, however, that personal mentoring from senior members in joint research projects also provided them with significant assistance as did the comments they received from editors and reviewers who responded to the manuscripts they submitted for publication.

It would not be unreasonable to speculate that purposeful reading, writing, and feedback situated within a genuine professional research context plays a far more influential role in the development of a computer scientist's writing skills than a single course on writing instruction. At least none of the computer scientists who responded to the surveys claimed that the writing courses they took as undergraduates were more valuable to them than the input they received from other sources. Unless a course is highly relevant to the specific language needs of the learners and the context of their discipline, it may be argued that the course has limited value. This, of course, will be suggested in this study, but more rigorous investigation will be required to confirm this.

3.4.4 Time

In the preliminary surveys, short interviews, and casual conversations, respondents stated that time was an important element that affected their writing in two important ways: the selection of research projects and the process selected for writing papers.

Concerning research, several faculty members stated that different projects required different time-frames. Serious and substantial research projects required several years. Some said anywhere from three to seven. Less significant projects required less time. The balance between quality and quantity was not always easy to achieve. There were other factors to be considered, too.

For example, there was strong pressure at the local level to produce results and publish them annually, particularly if the faculty member was hired to do research full-time and did not have to teach any courses. This could be responded to with short and/or progressive research where results could be generated and written up quickly. There was also strong pressure within the discipline to obtain results and get them published before competing researchers "got there first." This might also suggest a quick, pragmatic approach. On the other hand, it was the substantial findings and/or applications that built reputations fastest and brought greater fame (and ultimately money) to the university or research institution. There was also the complication of tenure and/or future career plans. Since the accomplishments of all faculty would be reviewed and tenure decisions made at the end of three years (for the vast majority, March 1996) there was a great deal of calculation on the part of faculty to select projects that could be completed within that time frame. In addition, some of the foreign faculty had no plans to remain in Japan until retirement. These researchers, as well, carefully selected work that could be completed within their personal employment time frame, would enable them to build their CVs, and would increase their chances at landing a good job at the next career move. All these issues had to be considered and balanced before many of the local computer scientists could begin to chart their research agenda. The resulting research decisions, of course, greatly affected the number and quality of papers each faculty member or team would write.

The other, but related, issue concerning time involved the production of individual papers. Here the abundance or scarcity of time determined how much attention could be devoted to each step of the research-writing process. When deadlines were tight (self-imposed, administration imposed, conference committee imposed, editor imposed, etc.), some shortcuts had to be taken. Though few commented in detail about specific strategies in the initial investigation, there was occasionally a casual negative remark about a colleague who skillfully used others to assist in the rapid production of a particular publication or paper. There were obviously some unwritten ethics about research and writing that some of the computer scientists followed and others didn't. One casual observation, however, was that the issue of ethics appeared to vary from nationality to nationality (though it could not be confirmed). At least in casual conversations over meals, members of one nationality would often make generalizations about the ethics of colleagues of other nationalities. It was hard to determine, however, if these opinions represented fact. Did the differences in how one resolved these issues of time depend more on one's nationality or did they depend on each individual's personality and value system? Since computer science is an international discipline, how do differences in opinion on "proper and professional" behavior get resolved? This would also be worthy of future study.

3.4.5 Timeliness

Another issue of importance in the context of writing concerns timing. The popularity of various areas of research or specialization seems to change rapidly in computer science, perhaps more rapidly than in other fields. Since most computer scientists conduct research and compete for the opportunity to present their results at a conference or publish them in a journal with a severe limit on available slots, some respondents commented that it is not only the significance of results that determines acceptance by the profession's *gatekeepers*, but it is also the timeliness of the topic. To quote one respondent, "What is hot one year may be ancient history the next." Keeping abreast of new developments and predicting what research will *sell* in upcoming issues of a target journal or conference seems to be a necessary part of a computer scientist's work. This, naturally, affects writing. However, the specifics on *how* each writer "marketed" his research could not be determined during the preliminary phase of investigation. This would require more research later.

3.4.6 Authorship and Acknowledgements

Another interesting feature of professional writing that surfaced from the survey and interview data was the issue of authorship and acknowledgement. In other words, who gets credit for a document and its contents? Respondents claimed they considered the following questions in their decision-making process.

What Names?

1) Who participated in the research and/or writing?

2) How much time and/or energy did each member contribute?

3) What was the value of each member's contribution to the total success of the project?

4) What is the recognition value of each member's name among the intended readers of the document?

5) Are names on this document important for accomplishing the document's intended purpose(s)?

6) If the answer to question 5 is *yes*, then how can answers to questions 1 through 4 be used to the greatest advantage?

7) Are there other names that should be added to the list to show respect and/or appreciation?

8) Are there other names that should be added to the list to increase the document's value (particularly if the document will be refereed and the names will *not* be kept hidden)?

9) Are there other names that should be added to the list in order to assist friends, colleagues, or superiors lengthen their list of publications when current demands for publishing exceed a person's capacity to produce them under present circumstances?

10) Which names should be given credit in the author location and which names should be given credit in the acknowledgement location?

What Order?

1) List names in alphabetical order?

2) List names according to amount of participation?

3) List names according to the value of participation?

4) List names according to name value?

In the initial phase of questioning, it appeared that most of the faculty made a variety of decisions on the issue of giving credit for a particular piece of research. These differed according to the specific circumstances surrounding each publication. Few made the same decision every time. It was clear, however, that many of the respondents believed that some of the options listed above were dishonest and unprofessional in nature and that they would not chose them personally. Though superficial observation seemed to suggest that preferences could possibly be predicted according to nationality, deeper investigation seemed to suggest that it might rather be a matter of personal morals and/or level of professional competence which made socio-political considerations part of the writing and publication process. Though the cultural aspect of this topic will not be addressed in detail in this particular dissertation, it clearly would be excellent for future research.

3.4.7 Additional Miscellaneous Elements

In addition to the elements identified and briefly discussed above, the following items are also worthy of inclusion in a model of the computer science writing context due to their obvious importance in most every composing process. Since these elements pervade nearly all the professional literature in the field of composition, specific citations are only included when a particular element has received special attention that is of specific relevance to this study.

- the intended purpose of each writing episode

- the target audience for each written product (Porter, 1992)

  Porter offers a heuristic for forum analysis that is useful for thinking about the audience of a specific journal or conference. It includes all four of the traditional elements of communication: the writer, the reader, the message, and the medium.

- research experience and expertise

- research facilities, equipment, and funding

- knowledge of writing options (Swales, 1990; Bhatia, 1993; and Berkenkotter & Huckin, 1995)

  All four authors address the necessity of writers in professional fields to expand their repertoire of grammar and genre options that are sanctioned within their profession.

- the writer's personal memory (Bereiter & Sardamalia, 1987)

  These researchers include memory in both of their models of the writing process. As cognitive scientists, they recognize that memory plays a major role in the construction of texts.

3.5 Summary

In Chapter 3, several major elements of the computer science writing context have been identified. For review, the items are listed below.

- writing activities
- research
- reading
- previous writing instruction
- time
- timeliness
- authorship and acknowledgements
- writing purposes
- target audiences
- research experience and expertise

- research facilities, equipment, and funding
- knowledge of writing options
- personal memory

A simple list of activities and concerns that are important to computer scientists is not much help to educators concerned with developing appropriate writing support for the profession. How these elements are tied to each other and which activities proceed other activities must be outlined or else the flow of professional activity will not be evident. This was the goal of research conducted in Chapter 4.

Chapter 4

Constructing a Preliminary Model

The first phase of research was designed to gather information from both computer scientists and from relevant research in composition or related fields to begin to identify the major elements of the professional computer science writing context. These elements have been listed in Chapter 3. The next phase of research involved organizing these elements into a general schemata of professional activity so that important relationships could be recognized and that data concerning finer aspects of professional writing could be appropriately placed. More specifically, this research phase involved identifying an organizing principle to group and sequence the elements, placing the elements in a preliminary flow chart of professional thought and activity, and adding important details after further study of the initial computer science input.

4.1 Grouping the Writing Elements

Since writing processes and products must be the focal points in a model of professional writing practices, the first step in this second research phase was to find a suitable organizing principle for grouping and studying the writing that respondents identified as typical within the field of computer science. A review of the preliminary responses and further conversations with the computer faculty revealed that the words *information* and *resources* were, likely, two of the most important terms in the field of computers and that writing was most valued for its role in generating, obtaining, managing, and dispensing these two commodities. Since a model of writing in a discipline must reflect that discipline's world view, it was decided that *writing as a tool for the manipulation of information and resources* would be a more appropriate perspective from which to view computer science writing than any other organizing principle that might be imposed upon the model from other disciplines.[1] It also seemed appropriate that the four primary components of a computer system (*input, storage, processing, and output*) serve as part of the organizing metaphor for the four writing functions characteristic of computer science. Consequently, the first step in ordering the elements in a computer science context was to group writing according to

1) its role in storing information, i.e., **Storage-Directed Writing**;

2) its role in generating information, i.e., **Process-Directed Writing**;

3) its role in obtaining information or resources, i.e., **Input-Directed Writing**; and

4) its role in dispersing information, i.e., **Output-Directed Writing**.

The writing that fits under these categories is discussed in the following four sections.

[1]Kinneavy (1971) has proposed that within the field of English, each area of study takes a different perspective of writing. Linguistics frequently organizes writing around concepts of *grammar*; rhetoric, around *audience*; literature, around *readers*. Other fields organize writing by other value systems. Cognitive psychologists may organize writing according to *thinking patterns*; anthropologists, according to *believability* (Gertz, 1988); and political scientists according to issues of *power*. A perspective of writing within the field of computer science would only be distorted if the researcher imposed a value system other than the one used most by computer scientists.

4.1.1 Storage-Directed Writing

When computer scientists in this study were asked about their personal writing practices, it became apparent that one category of writing common to all was that which was intended to manage the tremendous inflow of information so that it could be remembered and used with the least amount of time and effort. This information came to them through reading various hard and electronic texts, through listening to others in private conversations or at professional forums, and through personal experience in and out of the research laboratory. Most of the computer faculty claimed that their need to "keep up with what's new" in their area of specialization and in related areas was crucial to their success as a computer scientist. The creation of new information in the field, however, occurred at an extremely rapid pace—almost too fast for faculty members to keep up, particularly the NNSs.[2] This required not only searching for efficient means to access information as quickly and easily as possible, but also it required them to manage this tremendous influx of data in such a way that it would not get lost. Since human memory has storage and recall limitations, most of the computer scientists in the sample employed various forms of writing to record that which they deemed worthy of remembering and in a form that made re-access most convenient. It was clearly writing for purposes of information storage.

Writing under this category primarily takes two forms: *notational support* and *organizational support.*

Notational Support

Notational support usually takes the form of notes in the margins of texts, notes and

[2]Non-Native Speakers of English

diagrams on scrap pieces of paper or in easily portable notebooks, or as notes written and stored electronically in a laptop or desktop computer depending upon the site where the information was first encountered. This writing is fairly impromptu with the simple purpose of recording a fact, a thought or an idea as quickly as possible to avoid losing it. Once the information is no longer needed, it is simply discarded, deleted, or occasionally lost. Often, however, this information is used as the basis for the second type of **Storage-Directed Writing**, labeled *organizational support.*

Organizational Support

Organizational support is primarily managerial in function; it is writing employed to process and structure the inflow of information so that it can be more easily understood and more efficiently recalled. This type of writing also requires more thought and effort to produce. The two most common storage devices used by the University of Aizu faculty were electronic files and hard copy notebooks. The purpose of organizational support writing is to organize information in such a way that it can easily be found again and incorporated quickly into the other three categories of writing: **Process-Directed**, **Input-Directed**, and **Output-Directed Writing**. Some filed information in computer files by topic, others kept running bibliographies with annotations, and others kept a series of notebooks organized by topic, by date, or by source of input (e.g., name of journal). One streamlined the writing process by nearly making writing and/or typing almost unnecessary. This professor simply hired students to regularly scan in the data he had highlighted in the articles he needed to access at future times. Generally, the computer was the tool most commonly employed for producing writing that was organizational in nature, particularly since information could be quickly cut and pasted from these files into various other documents when needed without having to retype anything.

One final note about **Storage-Directed Writing** is that many faculty members claimed to employ a variety of different strategies that changed according to the circumstances at the time. It was also observed that some of the computer scientists interviewed had established regular patterns of handling large quantities of information, and others were in an experimental stage, not knowing which system worked best for them. One topic for future study would be to look at this issue in more detail to see how personality, background culture, and experience in the field affect the selection of **Storage-Directed Writing** strategies and how these strategies affect the speed and quality of research and publication.

4.1.2 Process-Directed Writing

Process-Directed writing is primarily writing employed to aid the construction of **Input-Directed** and **Output-Directed** texts. It is writing that often occurs on the back of recycled printer paper or in notebooks for the purpose of thinking through problems, organizing ideas and solutions, and experimenting with potential outlines for documents. It is generally used along with a generous amount of diagraming and mathematical calculation. Since most documents are generated on the computer, **Process-Directed Writing** occurred prior to and parallel with the writing that actually takes place on the computer screen. Some faculty faithfully held on to these records, and others disposed of them as soon as the final product was complete and saved in computer memory. None of the faculty who responded to initial inquiries claimed to write entirely on the screen without ever using some sort of writing on paper to assist him or her think. Though computer technology is beginning to offer better electronic note pads for easier electronic doodling, paper and pencil have yet to be replaced, even in computer science.

4.1.3 Input-Directed Writing

Writing activity under this category is generally persuasive in nature and includes all texts written for purposes of obtaining information, approval, human resources, financial resources, or equipment and facility resources. Because **Input-Directed** and **Output-Directed** writing are central to the work of computer science, specific examples have been listed along with brief descriptions. In the computer science field, **Input-Directed** writing includes the following according to those surveyed in this study.

Type: stand-alone questions and/or requests
Medium: usually sent by e-mail
Intended Purpose: to obtain a particular piece of information or an entire document of information.
Additional Comments: The context for the exchange is usually familiar to all parties and has sufficient precedence within the computer science community. The language is generally straightforward designed to achieve its aims with the minimum investment of time. Sometimes the requests are automated; they are processed by computer servers which require specific commands (e.g., GET filename) and deliver programmed results.

Type: textually embedded questions and/or requests
Medium: usually sent by e-mail and/or hard copy
intended Purpose: to obtain a particular piece of information or an entire document of information.

Additional Comments: This document is often letter-like (i.e. has an opening, body, and closing). The context may be unfamiliar to the reader, and thus requires greater backgrounding; or the reader may occupy a higher status in the discipline, and thus require questions and requests couched in *official* form with a *respectful* ethos.

Type: proposals

Medium: usually sent by hard copy and occasionally by e-mail

Intended Purpose: to obtain approval and/or resources to carry out a particular project or to present information in a public forum.

Additional Comments: The form and content of proposals are usually governed by the person or group possessing the authority of approval and the desired resources. Many features of the form in which they appear have been borrowed from other examples inside and outside the computer science discipline and been adapted to meet specific local needs.

Type: course examinations, tests, exercises, problems

Medium: hard copy and electronic

Intended Purpose: to collect information on student achievement.

Additional Comments: This writing is educational in nature. It is designed to test one's knowledge of a particular subject or one's ability to carry out a particular task. The feedback comes from learners, primarily in computer science courses.

Type: technical reports

Medium: usually, both hardcopy and electronic

Intended Purpose: to receive feedback from colleagues, generally in the same area of specialization, on preliminary research results.

Additional Comments: Technical reports are used for a variety of purposes in computer science, one of which is the solicitation of feedback, another of which is to claim research territory early. They generally are 15-50 pages in length, have a cover page, an abstract, an introduction, a review of related literature, a description of the research methods, a presentation of research results, and a discussion or conclusion section. There are often 10 or more references listed at the end. Few technical reports are without diagrams, graphs, or mathematical equations.[3]

Type: general public relations material (brochures, annual reviews of department or university activities, alumni magazines, etc.)
Medium: slick, colorfully illustrated hardcopy or Web pages
Intended Purpose: to construct a positive public image of a person or institution for purposes of recruiting students, faculty, research partners, or funding.
Additional Comments: As with technical reports, PR may be used for a variety of purposes. To build an ethos that will attract human and financial resources is one of them.

Type: announcements of professional events (e.g., conferences)
Medium: posters, journal pages, electronic pages
Intended Purpose: to generate participation by others, generally in the computer profession, in professional events or forums for the exchange of discipline-related information.
Additional Comments: These are usually straightforward, serious, and seldom contain humor of any form. Single-page postal mailings are usually the most text-oriented with little whitespace or graphic support. In addition to the conference theme and specific topics of concern, there is commonly a list of featured speakers and a long list of conference program members, serving in various capacities, followed

[3]Technical reports will be discussed in greater detail later in this dissertation.

by their university or professional affiliations. Apparently, names attract conference participants as much as, if not more than, presentation topics.

Type: editorial comments from colleagues, editors or referees

Medium: hard copy or electronic

Intended Purpose: to show error in information or in conventions of acceptable professional dissemination to an author so that he or she might improve a piece of writing before publication.

Additional Comments: Comments from referees are usually blind, i.e., the reader does not know the identity of the writer. In addition, professional computer science organizations, such as the IEEE and the ACM, have ethics statements governing this process to prevent papers from being approved according to criteria other than their content and quality of presentation.

Type: public letters from editors, SIG chairs, etc.

Medium: hardcopy or electronic

Intended Purpose: to promote a change in community thinking or behavior, particularly in regard to professional forums such as journals, newsletters, or conferences.

Additional Comments: These vary in tone from formal and impersonal to casual and personal, generally corresponding to the prestige of the publication or conference proceedings. The higher the prestige, the more formal the tone. Variations in this predominant pattern may be due to personality and/or nationality.

Type: book/software reviews

Medium: (usually) bound hardcopy

Intended Purpose: to encourage or discourage purchase/use.

Additional Comments: Reviews generally appear in special interest group journals or newsletters rather than in the top journals. They are frequently short, conversational in nature, and commonly use the first-person pronoun. Occasionally, they also may include humor.

4.1.4 Output-Directed Writing

The primary purpose of documents under this category is that of disseminating information to others, usually other members of the discipline and often within the same area of specialization. **Output-Directed Writing** takes two forms according to its two sub-purposes: *community-building writing*, which attempts to describe features of the continually evolving computer science discourse community, and *knowledge-building writing*, which attempts to continually revise the community's knowledge base by either adding new information to it or altering information that is already there. As the computer science field is primarily a knowledge-generating profession concerned with both the development and dissemination of computer-related theory and application, **Output-Directed Writing** figures centrally in accomplishing the discipline's goals. The professional writing that falls under this category and its two subdivisions can be described as follows.

Community-Building Writing

Community-building writing consists of texts that attempt to construct representations of various portions of the computer science discourse community concerning its structure and the members who inhabit it. Writing for this purpose includes the following.

Type: descriptions of universities, departments, labs, programs, projects, task forces, committees, etc.

Medium: usually hardcopy or Web pages

Intended Purpose: to identify groups of computer scientists who have united in some organizational form to accomplish some specific professional task.

Additional Comments: These descriptions usually appear along with public relations material, in conference announcements, or other writing. They seldom appear alone.

Type: biographical sketches

Medium: hardcopy or electronic

Intended Purpose: to identify specific members of the computer community and describe their work and their authority within the computer science hierarchy.

Additional Comments: These paragraphs usually appear at the end of articles, in books where descriptions of the authors are located, in conference proceedings, on personal Web pages, etc. They primarily accompany the dissemination of research-related work to inform the audience who is speaking and from what authority. Though biographical sketches are generally serious in nature, those on Web pages employ far greater use of humor, popular jargon, and personal information unrelated to computer science.

Type: public letters from an editor, SIG chair, etc.

Medium: (usually) bound hardcopy (in journal or SIG periodical)

Intended Purpose: to inform the readership of a new policy (i.e. change in the profession's structure) or a change in leadership (e.g., change in the hierarchy).

Additional Comments: These public announcements are perhaps the most important in disseminating news about the computer science community, particularly as it relates to the structure and governance of professional forums for exchange such as journals, newsletters, and conferences. They usually appear at the front of journals or SIG publications.

Knowledge-Building Writing

Knowledge-building writing is considered the most important by the computer science community surveyed because it is the writing under this category that demonstrates personal/institutional competence, personal/institutional accomplishment, and it builds the profession's knowledge base, the discipline's central goal. Consequently, much of this writing carries the greatest weight in evaluations for higher rank and benefits within the computer science profession, particularly in university settings. These documents are by far the most rule-governed and require the greatest amount of energy and time to produce.

Type: oral presentation aids
Medium: OHP transparencies, slides
Intended Purpose: to aide a listener's comprehension and memory.
Additional Comments: Oral presentation aids consist primarily of graphs, diagrams, mathematical equations, or programming code with brief textual support. Transparencies that contain only text usually consist of outlines or specific quotations.

Type: technical reports
Medium: hardcopy and electronic
Intended Purpose: to disseminate preliminary research results in order to claim research territory (also, used to supplement a request for feedback from other researchers in the same area of specialization).

Additional Comments: In addition to the description provided earlier on technical reports, these documents usually permit the author more freedom in content and presentation style than do journal articles since there are seldom many rules imposed upon their design from outside. They are generally published in-house by the university or research institution and are seldom restricted beyond cover design and paper size.

Type: conference papers/proceedings

Medium: bound hardcopy, electronic

Intended Purpose: to distribute research results in a short formal paper form with fewer restrictions and quicker turn around than for a journal publication.

Additional Comments: These often range in length from 4-10 pages and contain all the features of a journal article except for their extreme conciseness. References to other literature are particularly brief. Most of the attention is devoted to research results.

Type: article in SIG publications

Medium: bound hardcopy and/or electronic periodical

Intended Purpose: to disseminate research results to a narrow range of specialists in a timely manner under (usually) fewer restrictions than for a journal article.

Additional Comments: These are often sent camera-ready for publication as is. There is seldom much input from referees or editors concerning revision. There is also a wider variety of presentation styles than there is in refereed journals.

Type: journal articles

Medium: bound hardcopy and electronic on occasion

Intended Purpose: to formally announce significant research accomplishments.

Additional Comments: Three of the most common types of journal articles are referred to as "regular papers" (30-35 pages), "concise papers" (10-15 pages), and "correspondences" (4-15 pages). Most contain abstracts, introductions, discussions of related work by others, descriptions of research methodology, presentations and discussions results, and conclusions. They also have reference lists, acknowledgements, and biographical information about the authors.

Type: popular books (or portions of books)

Medium: bound hardcopy

Intended Purpose: to educate professionals and non-professionals who are interested in the subject matter.

Additional Comments: Since technology changes so rapidly, computer science books intended for general audiences appear quickly and seldom sell well for longer than two years, unless the material is more philosophical in nature than it is application-oriented.

Type: scholarly books

Medium: bound hardcopy

Intended Purpose: to educate other professionals and highly interested others.

Additional Comments: Scholarly books tend to consolidate research and present well-known theory that has been around for some time. Content material is seldom comprehensible to those outside a narrow range of specialization.

Type: textbooks

Medium: bound hardcopy

Intended Purpose: to educate novices (e.g., students).

Additional Comments: These generally fall into two categories. One kind of textbook is designed for computer science majors and the other is designed for students in other majors. Textbooks for computer science students frequently focus on one content area in the discipline and cover the material fairly deeply. Textbooks designed for non-computer science majors generally introduce the basic components of computers and show how computers are used in various fields. This second type of textbook contains many glossy photos of the latest technology and becomes outdated very quickly. Though written for non-computer science majors, these textbooks are frequently used in ESL/EFL instruction for computer science majors since they introduce computer science vocabulary quite effectively and can be easily understood by ESL/EFL teachers who may be unfamiliar with computer science.

Type: software/hardware documentation
Medium: hard copy and electronic manuals
Primary Purpose: to educate new users.
Additional Comments: Documentation varies widely in style and seldom serves novices well. There are often assumptions made by the authors concerning the reader's knowledge that are inaccurate. Consequently, there is a large percent of user-unfriendly instructions that accompanies various equipment and software.

4.2 Features of the Preliminary Model

Up to this point in the research, writing has been grouped according to four primary functions. Computer science writing can be **Storage-Directed, Process-Directed, Input-Directed**, and **Output-Directed**. A simple model showing the sequence of these writing activities might look like the following.

4.2.1 Model of Four Primary Writing Functions

(1) Input-Directed Writing

⇓ ⇓ ⇓

obtains information and resources

↓

(2) Storage-Directed Writing

⇓ ⇓ ⇓

organizes information for efficient access

↓

(3) Process-Directed Writing

⇓ ⇓ ⇓

parallels research activities to generate new information

↓

(4) Output-Directed Writing

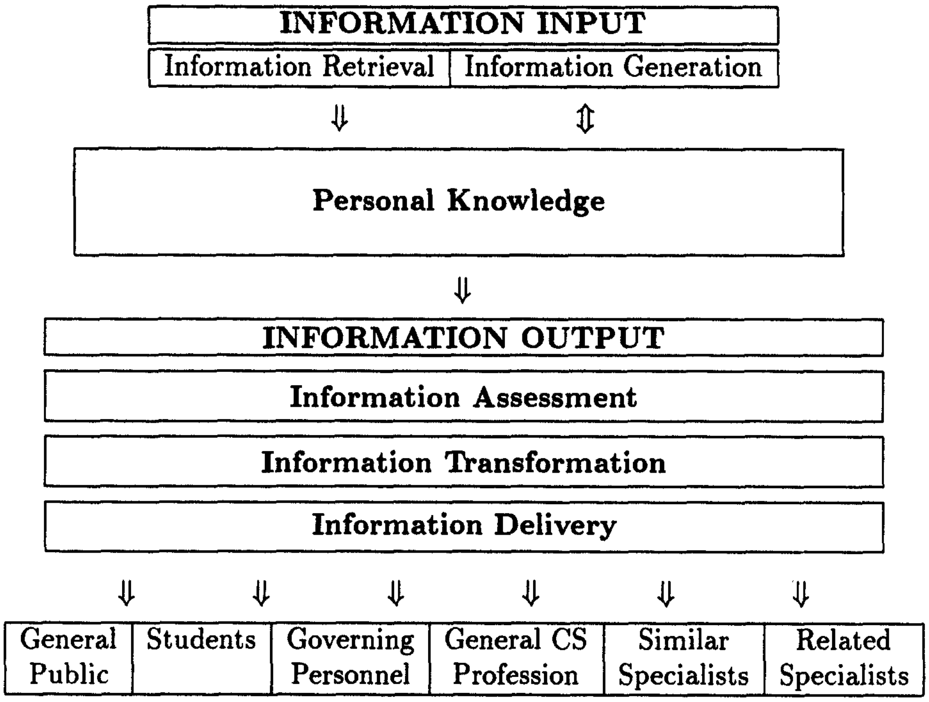
⇓ ⇓ ⇓

disseminates the newly generated information

Writers in the field of computer science employ **Input-Directed** writing to assist in gathering information; employ **Storage-Directed** writing to assist in organizing information; employ **Process-Directed** writing to assist in generating new information; and then employ **Output-Directed** writing to assist in disseminating that information. This flow of information through the system is clear and easy for novices just entering computer science to comprehend, but when other elements of the writing context are added (a frequent oversight in writing instruction), the picture gets more complicated. Information originates at various locations, is used for a variety of purposes, is targeted for different audiences, is processed and structured in multiple ways, and flows in cycles that are influenced by many contextual forces. A model of professional writing activities cannot stop here. A better way to model computer science writing practices would be to show the flow of specific thoughts and tasks a writer engages in. If writing is a mental and physical activity directed at accomplishing specific goals, both thought and activity must be represented in the model.

The following sections record attempts to do this. The first section gives a macro-perspective of professional practice in computer science, and the sections that follow offer micro-perspectives showing more detail. These are preliminary in nature and were constructed to lay the foundation for further research conducted in later chapters.

4.3 A Macro-Perspective



A macro-perspective of computer science practice shows that information primarily comes from two different sources. A computer scientist obtains information from others or from the result of personal research activities. This information builds personal knowledge or expertise. The flow of information is primarily from these two sources to personal knowledge; however, there is also a considerable amount of information that flows back into the research environment. As a matter of fact, one might say that the quantity and quality of ideas one can generate depends solely on the quantity and quality of information that flows into the system through Information Retrieval activities. In other words, a computer scientist cannot come up with a significant number of quality ideas without first having a hefty inflow of significant ideas from others.

The output phase is also important. As we discovered earlier, it is as important as the input phase. Here, significant knowledge is continually identified, targeted, and transformed to meet the expectations of one or more of six different audiences.

These are the most general aspects of professional practice. The model, of course, must include more detail. We need to see how writing, and all the other elements we have identified, fit into this picture. This will require several micro-perspectives.

4.4 Micro-Perspectives: INFO INPUT

Personal knowledge is increased in two ways: 1) by retrieving information from others and 2) through personal and/or group research. The following two diagrams illustrate these professional activities in more detail.

4.4.1 Information Retrieval Model

INFORMATION INPUT	
Identify Source	
⇓ *Written?* ⇓	⇓ *Spoken?* ⇓
journals technical reports conference proceedings etc.	seminars discussions conference presentations etc.
Read	Listen
Evaluate Quality of Information	
Is it personally useful?	
⇓ *yes* ⇓	⇓ *no* ⇓
Provide Notational Support	Store Passively in Memory
Evaluate Quantity of Information	
Is it manageable? Can information be found easily?	
⇓ *yes* ⇓	⇓ *no* ⇓
Continue Adding More	Provide Organizational Support
Add Information to Personal Knowledge	

⇓

Personal Knowledge

As revealed earlier in this chapter, computer scientists generally do a great deal of reading. They also frequent professional conferences, seminars, and often engage in dialog with peers via e-mail or face-to-face. A large inflow of information is necessary not only to build expertise but also to feed the knowledge generation process, a computer scientist's central concern.

Not everything that computer scientists hear or read is useful to their professional interests. That which has little value is stored passively in memory, and that deemed useful is marked, highlighted, or supplemented with notation for future use. Frequently, the amount of information becomes unmanageable. Information captured with *notational support* can be forgotten or lost. When this happens, many computer scientists employ various techniques to organize their collections of knowledge so that information can be remembered more easily and found more efficiently. The computer is the most popular device for long-term storage since quotations, equations, diagrams, references, and miscellaneous facts can be called up quickly and copied into other documents. *Organizational support* is writing that assists writers find previously read or heard information quickly and efficiently for rapid employment in the production of information output. This writing that occurs in the Information Retrieval phase of the model is **Input-Directed Writing**.

4.4.2 Information Generation Model

<table>
<tr><td colspan="2">INFORMATION INPUT</td></tr>
<tr><td colspan="2">Generate Idea for Research Project</td></tr>
<tr><td colspan="2">Evaluate Idea</td></tr>
<tr><td colspan="2">Will the research generate significant results?</td></tr>
<tr><td>⇓ yes ⇓</td><td>⇓ no ⇓</td></tr>
<tr><td>Proceed</td><td>⇑ Return to Generate Idea ⇑</td></tr>
<tr><td colspan="2">Evaluate Timing</td></tr>
<tr><td colspan="2">Will there be enough interest to publish/market these results?</td></tr>
<tr><td>⇓ yes ⇓</td><td>⇓ no ⇓</td></tr>
<tr><td>Proceed</td><td>⇑ Return to Generate Idea ⇑</td></tr>
<tr><td colspan="2">Evaluate Time</td></tr>
<tr><td colspan="2">Will the time required fit into my career schedule?
Will the time required fit other time schedules?</td></tr>
<tr><td>⇓ yes ⇓</td><td>⇓ no ⇓</td></tr>
<tr><td>Proceed</td><td>⇑ Return to Generate Idea ⇑</td></tr>
<tr><td colspan="2">Evaluate Human Resources</td></tr>
<tr><td colspan="2">Do I have enough expertise to carry out this project?</td></tr>
<tr><td>⇓ yes ⇓</td><td>⇓ no ⇓</td></tr>
<tr><td>Proceed</td><td>⇑ Return to Generate Idea ⇑
or
⇐ ⇐ Recruit Research Partners ⇐ ⇐</td></tr>
<tr><td colspan="2">Evaluate Physical Resources</td></tr>
<tr><td colspan="2">Do I have funds and/or equipment to carry out this project?</td></tr>
<tr><td>⇓ yes ⇓</td><td>⇓ no ⇓</td></tr>
<tr><td>Proceed</td><td>⇑ Return to Generate Idea ⇑
or
⇐ ⇐ Recruit Funds and/or Equipment ⇐ ⇐</td></tr>
<tr><td colspan="2">Make Research Plan</td></tr>
<tr><td colspan="2">Conduct Research</td></tr>
<tr><td colspan="2">Generate Results (New Information)</td></tr>
<tr><td colspan="2">Add Information to Personal Knowledge</td></tr>
</table>

⇓

Personal Knowledge

Information generation is the primary goal of computer science. In professional practice, computer scientists generate ideas, evaluate the potential results, evaluate the time and timing, evaluate the resources necessary to carry out the project, and finally make a research plan if all the signals are green. The research activities generate results which are fed into personal knowledge and then assessed in later stages to determine the potential for, and form of, dissemination. Writing often occurs in the research phase in the form of various notations, charts, graphs, equations, outlines, and whatever else is necessary to record findings and assist creative thinking. It is both **Input-Directed Writing** and **Process-Directed Writing**.

4.5 Micro-Perspectives: INFO OUTPUT

4.5.1 Information Assessment Model

Personal Knowledge

⇓

Information Assessment	
Evaluate Knowledge	
Do I possess some significant information?	
⇓ *yes* ⇓	⇓ *no* ⇓
Proceed	⇑ Return to **INFORMATION INPUT** ⇑

Select Audience		
Who will benefit from this information?		
General Public?	*Students?*	*Governing Personnel?*
General CS Profession?	*Similar Specialists?*	*Related Specialists?*

⇓

Information Transformation

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	Students	Governing Personnel	General CS Profession	Similar Specialists	Related Specialists

This diagram shows a little more detail of the **INFORMATION OUTPUT** half of the model. Computer scientists generally begin the OUTPUT phase by assessing the significance of some information they posses. The significance of a specific piece of information will not only determine which audience the information is suitable for, but it will also help them determine which genre is more appropriate for dissemination.

The **Information Transformation** phase is the sequence of activities that generally is of most interest to writing researchers and educators in EST (English for Science and Technology). Generally, information is transformed to conform to the specific expectations of six different audiences. Each of these is shown in more detail in the following six sub-models.

4.5.2 Info Transformation ⇒ General Public

Information Transformation	
Medium Selection	
Which medium is appropriate?	
⇓ *Spoken?* ⇓	⇓ *Written?* ⇓
Genre Selection	
Which genre is appropriate?	
⇓ *Spoken Genres?* ⇓	⇓ *Written Genres?* ⇓
speech	speech text or notes
lecture	lecture notes, OHP transparencies
seminar	seminar notes, OHP transparencies, handouts
workshop	workshop notes, OHP transparencies, handouts
	magazine article
	brochure
	instruction manual
	electronic documentation
	book (or part of a book)
Write Document	
Determine Authors and Acknowledgements	
Obtain as much Feedback as Possible/Necessary	
Revise Document as Often as Possible/Necessary	
Disseminate to Target Audience	

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	Students	Governing Personnel	General CS Profession	Similar Specialists	Related Specialists

Information that is targeted for the general public takes two forms: spoken and written. The spoken genres that are frequently employed by computer scientists are public speeches, lectures, seminars, and workshops. These are generally supported with written genres, such as various forms of notes to facilitate the talk, OHP transparencies to illustrate concepts and/or factual information for the audience, and handouts to either guide the audience through the presentation or to provide more information that can be read later at leisure. On occasion, the distinctions between the spoken genres blur. Computer science speeches are primarily to inform and entertain large audiences. Lectures are primarily to inform and entertain smaller, more serious, audiences. Seminars and workshops, on the other hand, are solely for serious study purposes with workshops tending to have strong hands-on audience participation (e.g. walking users through a particular activity performed on a computer). There are no hard lines between these genres however, for audience size and the intensity of interest are too vague to provide adequate characterizations of these spoken forms.

The written genres that occur independently of spoken forms include popular magazine and newsletter articles, often supported with state-of-the-art visuals, a heavy dose of popular computer slang, and a fair amount of humor. There are also brochures for publicity purposes as well as instruction manuals and electronic documentation to instruct general users how to operate software or hardware. Computer scientists also write books, on occasion. If they are for the general public, they are frequently general information guides for novices. Books on how to make the best use of Internet and World Wide Web are currently the most popular.

Once the genre is selected, it is then written or prepared, authors and acknowledgements are determined, feedback is obtained, revisions are made, and the final product is then delivered to its intended audience.

4.5.3 Info Transformation ⇒ Students

Information Transformation	
Medium Selection	
Which medium is appropriate?	
Spoken?	*Written?*
Genre Selection	
Which genre is appropriate?	
⇓ *Spoken Genres?* ⇓	⇓ *Written Genres?* ⇓
speech	speech text or notes
lecture	lecture notes, OHP transparencies
seminar	seminar notes, OHP transparencies, handouts
workshop	workshop notes, OHP transparencies, handouts
	practice exercises/problems
	computerized tutorials
	quiz or test
	midterm or final examination
	entrance examination (particularly in Asia)
	instruction manual
	electronic documentation
	textbook (or part of a textbook)
Write Document	
Determine Authors and Acknowledgements	
Obtain as much Feedback as Possible/Necessary	
Revise Document as Often as Possible/Necessary	
Disseminate to Target Audience	

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	**Students**	Governing Personnel	General CS Profession	Similar Specialists	Related Specialists

The obvious goal of information directed at students is education—specifically, their orientation to the content and culture of computer science. Some information is spoken and some is written; some genres are for purposes of giving information (content instruction) and some are for purposes of receiving information (comprehension assessment).

Since computer science is heavily application oriented, perhaps 50% or more of the education takes place in front of a computer. The most common educational methodology employed, at least at the University of Aizu, is a presentation mode where faculty employ OHP transparencies or computer screen displays projected on OHP screens to offer instruction while each student sits in front of a computer, listening to the information and then applying it at his machine. There are often student teaching assistants who assist students in the manipulation of computer commands, particularly at the early stages of their instruction.

Other methods of instruction imitate the norms of information exchange that take place between computer science professionals. There are frequent lectures and seminars, small group discussions and research activities, all for the purpose of *apprenticing* novices into the thinking, research procedures, and writing of the computer profession. Since much of the research (and education) is carried out in groups, collaborative writing (both in the preparation of instruction materials and in the preparation of research papers) is quite common. In fact, few professors or students create documents without significant input at the feedback stage and without significant revision before delivering the final product to its intended audience.

4.5.4 Info Transformation ⇒ Governing Personnel

Information Transformation		
Request Analysis		
What information has been requested?		
In what medium? Spoken or written?		
In what official format?		
Who will read it?		
What are the readers' unwritten expectations?		
How will this affect the following?		
selection of content	selection of language	selection of format
Write Document		
Determine Authors and Acknowledgements		
Obtain as much Feedback as Possible/Necessary		
Revise Document as Often as Possible/Necessary		
Disseminate to Target Audience		

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	Students	**Governing Personnel**	General CS Profession	Similar Specialists	Related Specialists

Written (or spoken) information for this particular audience is highly dependent upon the local environment. The content and form of the information is determined by 1) official regulations that usually state explicitly what information is required in what form and 2) by the personalities of those who will read the information. Since most writing of this variety is submitted to those in power for purposes of obtaining approval and/or funding, to demonstrate progress and/or accomplishments, or to obtain salary increases and/or promotions,[4] great care is devoted to these documents.

[4] In Japan, writing may be used to influence promotion decisions at a university (e.g., a move from Assistant Professor to Associate Professor) and the higher salary scale that automatically accompanies that rank, but for the most part, salaries (and rank) are awarded according to age and/or years of experience. This is based upon the assumption that the older one gets and/or the more experience one accumulates, the higher the individual's expertise and value to the institution. The Japanese university system does not base compensation on *actual* skill or competence. Thus, all 60-year-old professors will earn more than 50-year-old professors irrespective of *actual* ability. Consequently, writing plays much less of a role in promotion and salary decisions in Japan than in the United States.

The situation differs, however, if the writing is mere formality and decisions are based upon other criteria. In this case, much less time and effort is devoted to constructing the requested information.

It is interesting to note that in a rapidly evolving field such as computer science, it is not unusual for younger faculty to possess higher computer competence due to increasingly better computer science curriculums and higher energy levels that younger researchers can devote to continual study. This phenomenon obviously complicates a system where rank and salary are based on age and length (not quality) of experience. This affects professional writing in the sense that highly talented young researchers are occasionally under obligation (sometimes even pressure) to share credit for research results, and the accompanying papers (technical reports, journal articles, etc.), with their superiors who can no longer keep up with developments in the field. To what extent this phenomenon is also present in the United States or other western nations is a matter for further investigation.

4.5.5 Info Transformation ⇒ General CS Profession

Information Transformation	
Significance Ranking	
How significant is this knowledge/product to the CS profession?	
⇓ *Highly significant?* ⇓	⇓ *Moderately significant* ⇓
able to compete for major dissemination via the most discriminating professional avenues	able to compete for minor dissemination via less discriminating professional avenues
Medium Selection	
Which medium is appropriate?	
Spoken?	*Written?*
Genre Selection	
Which genre is appropriate?	
⇓ *Spoken Genres?* ⇓	⇓ *Written Genres?* ⇓
keynote speech	speech text or notes
conference presentation	presentation notes, OHP transparencies
lecture	lecture notes, OHP transparencies
seminar	seminar notes, OHP transparencies, handouts
workshop	workshop notes, OHP transparencies, handouts
	general correspondence
	instruction manual
	electronic documentation
	technical report
	SIG newsletter or SIG journal article
	review of information/product
	major journal article
	book (or part of a book)
Write Document	
Determine Authors and Acknowledgements	
Obtain as much Feedback as Possible/Necessary	
Revise Document as Often as Possible/Necessary	
Disseminate to Target Audience	

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	Students	Governing Personnel	**General CS Profession**	Similar Specialists	Related Specialists

Writing for the general computer science profession will be addressed in more detail later; however, a few matters may be appropriately discussed here. For example, *significance* appears to be a major concern to computer science researchers. Is the information they create and the resulting applications (software products, etc.) of significant interest and use to others in the computer profession? If their personal assessment of research results ranks the information fairly high, they will attempt to channel the material through the most competitive conferences, journals, or book publishers. Since there is a vast quantity of information generated in research laboratories around the world on a daily basis, and professionals can physically devote only so much time to reading this information, there is severe competition to disseminate information through channels that are read by the most people AND have earned reputations for passing on information of high significance.

Entry into premiere conferences and journals is limited by editors and referees who evaluate the quality of both the information and the mode of presentation via official *Referee Reports.* Samples of these may be found in Appendix C. Entry into the top publishing houses is determined by corporate policy usually based on the information's marketability. It is getting past the *gatekeepers*, who determine entry to conferences and journals, that concerns computer scientists the most.

Once a paper has been reviewed by experts in the same area of specialization, articles in major journals often go through several phases of revision. These revisions are based upon advice from both the referees and the editors. For published conference proceedings or less competitive journals, the feedback and revision cycle is less frequent, and occasionally non-existent. When the product is complete, the presentation is given or the article is published. Major journals may take a year or more to actually get the information in print. This also complicates the process since information can become obsolete quite rapidly in the computer field.

It is interesting to note that both highly significant information and moderately significant information can be disseminated generally via the same genres. Differences are primarily matters of competitiveness and convention. Less competitive avenues commonly offer more tolerance for creativity in presentation than do competitive ones.

4.5.6 Info Transformation $\Rightarrow$ Similar Specialists

Information Transformation	
Significance Ranking	
How significant is this knowledge/product to specialists working in same area?	
$\Downarrow$ *Highly significant?* $\Downarrow$	$\Downarrow$ *Moderately significant* $\Downarrow$
able to compete for major dissemination via the most discriminating professional avenues	able to compete for minor dissemination via less discriminating professional avenues
What are my OUTPUT aims?	
$\Downarrow$ *Build CS profession?* $\Downarrow$	$\Downarrow$ *Build CS knowledge base?* $\Downarrow$
What are my INPUT aims?	
Obtain information (e.g. feedback)?	*Obtain resources (e.g. researcher partners)?*
Medium Selection	
Which medium is appropriate?	
Spoken?	*Written?*
Genre Selection	
Which genre is appropriate?	
$\Downarrow$ *Spoken Genres?* $\Downarrow$	$\Downarrow$ *Written Genres?* $\Downarrow$
keynote speech	speech text or notes
conference presentation	presentation notes, OHP transparencies
lecture	lecture notes, OHP transparencies
seminar	seminar notes, OHP transparencies, handouts
workshop	workshop notes, OHP transparencies, handouts
	general correspondence
	instruction manual
	electronic documentation
	technical report
	letter to/from editor or SIG chair
	SIG newsletter or SIG journal article
	review of information/product
	major journal article
	book (or part of a book)
Write Document	
Determine Authors and Acknowledgements	
Obtain as much Feedback as Possible/Necessary	
Revise Document as Often as Possible/Necessary	
Disseminate to Target Audience	

Information Delivery

$\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$ $\Downarrow$

General Public	Students	Governing Personnel	General CS Profession	**Similar Specialists**	Related Specialists

Information targeted at computer science specialists in the same area generally follow the same procedure and operate according to the same concerns as information targeted at the general computer science community, with one major exception: The writer frequently writes for other specialists to obtain something in return. The most common INPUT desired by the writer is feedback on the current stage of research in a series of stages. Since completion of the project will yield results that may be significant enough for dissemination in the most competitive professional avenues, it is important to continually solicit feedback from experts during the process to increase the likelihood of impressive results. The genres most conducive to this are the technical report and conference papers/proceedings, though other genres may be employed as well.

The other concern that motivates some of the writing directed specifically at specialists in the same area is the desire for research partners to cooperate in joint research ventures. Most computer science research requires a great deal of money to purchase sophisticated equipment that needs regular updating. It also requires a broad range of skill and a great depth of knowledge that few individuals possess. In fact, there are few, if any, university or research institutions that possess all the research equipment and experienced staff necessary to carry out every research project envisioned. A single letter sent at the time of need is not enough. Computer scientists need to regularly disseminate their research to begin making contact with others who share similar research goals and who might "come in handy" when a staffing or equipment need arises. Sending technical reports to potential partners, presenting at conferences to attract potential partners, and publishing in SIG periodicals to identify oneself with a particular area of expertise appear to be frequent practices in computer science.

4.5.7 Info Transformation ⇒ Related Specialists

Information Transformation	
Significance Ranking	
How significant is this knowledge/product to specialists working in a related area?	
⇓ *Highly significant?* ⇓	⇓ *Moderately significant* ⇓
able to compete for major dissemination via the most discriminating professional avenues	able to compete for minor dissemination via less discriminating professional avenues

What are my OUTPUT aims?
Build another profession's knowledge?

What are my INPUT aims?	
Obtain information (e.g. feedback)?	*Obtain resources (e.g. researcher partners)?*

Medium Selection	
Which medium is appropriate?	
Spoken?	*Written?*

Genre Selection
Which genre is appropriate?

⇓ *Spoken Genres?* ⇓	⇓ *Written Genres?* ⇓
keynote speech	speech text or notes
conference presentation	presentation notes, OHP transparencies
lecture	lecture notes, OHP transparencies
seminar	seminar notes, OHP transparencies, handouts
workshop	workshop notes, OHP transparencies, handouts
	instruction manual
	electronic documentation
	general correspondence
	technical report
	SIG newsletter or SIG journal article
	review of information/product
	major journal article
	book (or part of a book)

Write Document
Determine Authors and Acknowledgements
Obtain as much Feedback as Possible/Necessary
Revise Document as Often as Possible/Necessary
Disseminate to Target Audience

Information Delivery

⇓ ⇓ ⇓ ⇓ ⇓ ⇓

General Public	Students	Governing Personnel	General CS Profession	Similar Specialists	**Related Specialists**

Writing directed at specialists in other areas, perhaps even in other disciplines, is most often conducted for the same purposes as that for specialists in the same area. There are input goals and output goals. The input goal is usually to gather information and cooperation from others frequently outside the computer profession. Since computer knowledge and applications apply to all disciplines, research work on a particular project (e.g., software to manage hospital records) typically requires expertise in the specific field of application (e.g., hospital staff and doctors).

A common output goal is to share information that might have *moderate significance* in computer science with professionals in other disciplines where the same information may take on a *highly significant* status. With professional accomplishment frequently measured by the number of publications in refereed journals or papers delivered at competitive conferences, it is common practice to search for dissemination avenues where information will earn the highest credit.

4.6 The Next Research Phase

The preliminary working models above reveal where writing is situated within the professional context of computer science. The models also show what major elements affect the retrieval and creation of information, the storage and organization of information, the assessment and targeting of information, as well as the transformation and dissemination of information. These form the foundation of any serious study of writing in a professional context. Until all the major writing activities can be identified and placed in their proper professional contexts, further investigation will yield increasingly fragile results with greater potential for erroneous conclusions. A serious study of professional writing, however, must go further still. It must uncover more specifics about each genre and how genres are constructed in the variety of situations that frequently present themselves. The best way to move deeper into the finer workings of professional writing practices at the next stage involves prolonged discussion with the professional writers who create these texts. That is the intent of Chapters 5 and 6.

The next phase of research involves case studies of two individual writers as they go about their professional work and naturally engage in writing throughout the process. The first writer is a native English writer born and raised in the United States; the second, a non-native English writer born and raised in Korea. Chapter 5 details the reflections, experiences, and practices of the American; and Chapter 6 details that of the Korean.

Chapter 5

Case Study: Native English Writer

5.1 Methodology

5.1.1 Selection of the Subject

In the initial phase of inquiry, several different questionnaires and surveys were distributed to faculty members over a period of two years. There were also micro-interviews of faculty (some recorded on cassette tape, others recorded with notations in notebooks) over a period of two and one-half years. The information collected provided the material for constructing the preliminary model of writing activities situated within the professional activities of the local computer science community. It also provided material for proposals for future research and educational applications outlined in Chapter 7.

One further objective of the faculty questionnaires was to identify active native and non-native English writers within the University of Aizu computer science faculty that would serve as suitable subjects for detailed case studies. Investigation of writing on a micro-scale was necessary to begin adding meat to the contextual skeleton of professional writing episodes in order to begin understanding what motivated these activities as well as understand the strategies writers employed to accomplish the goals for which the writing was intended.

The first case study would involve a native English writer. This would permit the researcher to begin identifying finer elements in the professional writing context, and it would serve as a baseline from which to construct the inquiries of other subjects. If a researcher is to remain objective in his or her investigations of writing practices within a particular profession, it is best not to impose preconceived ideas about what is ideal or what is normal. Rather, it is best to build hypotheses from the objective records of input from insiders within the field under study and to make value judgements only when there is sufficient criteria and a suitable volume of data to adequately do so. Consequently, the only criteria that were necessary for selecting the first subject for this case study investigation were the following.

1. The subject was a native English writer.

2. The subject was male.[1]

3. The subject was active in research and writing.

4. The subject was highly conscious of his writing.

5. The subject was enthusiastic about participating in this study.

After reviewing responses from the local computer science faculty and speaking with most of them informally, Dr. Ted Billard, Assistant Professor in the Department

[1]Over 90% of the local computer science faculty is male, roughly the same percentage as that found in other university and corporate computer research environments. It was, thus, believed wise to begin the study with two male writers and conduct research of female writers in future phases of the investigation beyond this present dissertation.

of Computer Software, appeared most suitable according to the five criteria listed above.

5.1.2 Method of Investigation

The multimodalic inquiry employed with the first subject involved the three phases of research detailed below:

Profiling the Subject

The first set of research activities was designed to construct a profile of the subject which would include general information about the subject's background, his research, and his writing. This profile would serve as a guide for identifying important aspects of professional practice that required further investigation and, thus, set the framework for questions asked during the recorded interviews. The profile was assembled in the following way.

1. The subject's responses were located in each computer file of faculty responses organized according to questionnaire number.
2. The responses were saved in a new computer file under the subject's name.
3. All responses were printed out and placed in a ringed notebook.
4. Data significant to this dissertation research were highlighted with markers and extracted from each questionnaire or survey for assembly into one profile constructed to present the information as coherently as possible.[2]

[2]If the researcher had been familiar with the computer science discourse community before this dissertation research began, he would have been able to construct one questionnaire that would

The Subject Profile that resulted from the first phase of study appears in Section 5.2.1 of the Results section of this chapter.

Investigating Personal Practices

Once the Subject Profile was constructed, this was read and compared with the information laid out in the preliminary models outlined in Chapter 4. Questions were then written, to guide personal interviews with the subject, and sent e-mail to allow the subject time to think about his responses. These questions may be found in Appendix C.[3]

The interview questions were designed primarily to solicit more detail about

- the subject's specific research goals and procedures,
- his writing during the Knowledge Generation stage,
- his writing and decision-making during the OUTPUT stages,
- details about the construction of specific documents, particularly the goals and means of accomplishing these goals in each section of the documents,
- details about the subject's writing education, particularly the input he received from his research and writing mentor,

gather this information in one simple response. This was not the case here, nor the case in most other EST investigations of professional disciplines. Consequently, the information was gathered in chunks from questionnaires targeted at collecting data on BOTH faculty writing practices, as well as on other matters related to education, that were necessary for informing the total English language program at the University of Aizu.

[3]It should be noted here that the interview questions were designed to initiate dialog on the topics relative to this study. They were general in nature so as not to impose any preconceptions on the subject being interviewed.

- details about the comments he receives from editors and reviewers and how he responds to their advice, and
- details about the subject's own responses to conference papers and manuscripts he reviews for journals.

A few days after the subject received the interview questions, the interviewer visited the subject's office, got a tour of the adjacent research laboratory the subject had just set up, was shown some educational software the subject and his lab team had developed for students, and then proceeded to the subject's office for a 2-hour interview. This was followed by another 2-hour interview a couple of days later.

During the interviews, the subject enthusiastically explained his work and writing practices, spoke about his writing training, and showed the interviewer several of his papers while explaining some of the context behind them. Some of the dialog followed the questions e-mailed earlier and some arose spontaneously from the general flow of discussion. These discussions were recorded on a cassette tape with a Sony TCM-AP1 Digital Pitch Control Cassette-Corder located on a large desk between the interviewer and the subject. After the interviews, these cassettes were transcribed by the interviewer and placed in the ringed-notebook along with the information from previous research.

The next step in research phase two involved the selection of important chunks of text from the transcripts, color coding significant statements according to topic with highlight markers,[4] and then analyzing these to identify important insights into the work, thinking, and writing of the representative computer scientist. This information

[4]This step was necessary because the interviewer permitted the subject to wander from topic to topic, only directing him enough to assure a wide coverage of topics and to clarify comments. This enabled the subject to follow his own train of thoughts in the order he believed best.

was then organized and added to the Subject Profile. This information appears in Section 5.2.2 of the Results section in this chapter.

Charting Ideas from Generation to Dissemination

In the third phase of research in this case study, the subject and researcher met again to look at and discuss the flow of some ideas through the cycle of professional activities outlined in Chapter 4. This involved following ideas from the generation stage through the research stage and on through the information transformation and dissemination stages. After some preliminary discussion about the subject's production of 4 documents from these ideas, the subject wrote his own description of the process he employed. This material appears in Section 5.2.3 of the Results section in this chapter. Copies of these 4 documents may be found in Appendix D.

5.2 Results

The results of this initial case study with the native speaker of English appear below. The first section presents the Subject Profile that was constructed in research phase one, the second section presents the interview with the subject conducted in research phase two, and the third section charts the flow of some related ideas from their genesis to four final products—results of research phase three. These three sections are followed by concluding observations and discussion in the concluding section of this chapter.

5.2.1 Subject Profile

Name: Edward A. Billard (Ted Billard)

Country of Birth: U.S.A.

Native Language: English

Education

1992 Ph.D. in Computer Science (University of California, San Diego)

1980 M.S. in Computer Science (University of Colorado, Boulder)

1973 B.S. Engineering Physics with Special Honors (University of Colorado, Boulder)

1973 B.S. Business (University of Colorado, Boulder)

University Rank: Assistant Professor

Department: Department of Computer Software

Research Lab: Operating Systems Laboratory (Director)

Area of Specialization: Autonomous Decentralized Systems

Professional Work Experience: Billard has had 8 years of experience in software engineering and management in the semiconductor field, 5 years of experience as a database consultant, 5 years of experience as a university researcher, and 6 years of experience as an educator.

Writing Education: At the undergraduate level, Billard had 5 semesters of writing instruction in literature courses. The 5th semester was an elective he selected because he enjoyed writing so much. At the graduate level, Billard learned writing from his research supervisor and from imitating other documents. Billard claimed that he was particularly fortunate to have a mentor who loved to write and provided Billard with a great deal of comment on his writing. When Billard began to submit papers to journals for publication, he began to receive feedback on his writing from reviewers and editors, mainly on the content. Only recently, with papers submitted in other disciplines, has he begun to receive specific feedback on his use of language.

Writing Production: Prior to employment at the University of Aizu, Billard wrote, presented, and published 4 conference papers; wrote and published 3 journal articles, and had 1 additional article going through the review process. Then during the first two years of employment at the University of Aizu (April 1, 1993, to April 1 1995) Billard significantly increased his dissemination rate with a total of 33 documents for peers in his field and in related fields. The documents consisted of 14 technical reports, 9 conference papers (presented, as well as published in proceedings), 2 SIG reports, and 8 journal articles (3 of which are forthcoming and 5 of which have been submitted and are currently going through the review process). In addition to this, he has written approximately 300 pages of miscellaneous documents for various other job-related purposes.

Target Channels of Dissemination: Billard stated that the following journals are most suitable for disseminating information about his research to the general computer science audience: *IEEE Transactions on Computers* and *IEEE Transactions on Software Engineering.* For specialized results, Billard says the following professional journals are ideal for disseminating his research results: *IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Automatic Control, IEEE Transactions on Parallel and Distributed Processing.* In addition, he stated that the most suitable conferences for disseminating his research results are the IEEE International Conference on Systems, Man, and Cybernetics (SMC); IEEE International Symposium on Intelligent Control (ISIC); IEEE International Symposium on Autonomous Decentralized Systems (ISADS); and ISCA International Conference on

Parallel and Distributed Computing Systems (PDCS).

Experience as a Referee: Billard has refereed journal papers for *IEEE Transactions on Computers* and *IEEE Transactions on Systems, Man, and Cybernetics.*

Writing Difficulties: Billard wrote in response to the questionnaires that the most difficult writing tasks for him were scientific papers (i.e., technical reports, journal articles, and conference papers)—in other words, the most formal and conventional documents designed for his academic peers.

Writing Expectations for His Students: In the initial questionnaires, Billard stated that he would expect his students (all of whom are native writers of Japanese) to write short five-page technical reports of experimental results and software designs which would possibly include an abstract, introduction, section on related work, a model, an analysis, experiments, a conclusion, and references. This would be supplemented with an oral presentation supported with OHP transparencies.

Instructional Expectations for the English Language-Teaching Faculty: Billard stated that he expected the CLR[5] to provide instruction in basic English, in writing short technical reports, and in making and using OHPs for oral presentations.

5.2.2 Subject Interview

The following sections contain excerpts from two interviews with Professor Ted Billard in December, 1994, and two additional interviews in June, 1995.

[5]Center for Language Research, the research and instructional support program serving students and nonnative English speakers at the University of Aizu

Discussion of the Computer Science Field

One of the first topics of discussion concerned the computer science field in general. The intent was to give the researcher a better grasp of how the field is defined and what common goals bind the community together. When asked to define the field, Billard pulled out a paraphrase of a definition issued by the Association for Computing Machinery (ACM) that he had found and was particularly fond of. It said,

> *Computer science is the systematic study of algorithmic*[6] *processes that describe and transform information—the theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all computer science is "What can be efficiently automated?"*

Billard commented on this by saying,

> Actually, this is a very good definition of computer science. I really like it....we use computers because we want to automate things, and we want to do it efficiently. If you can't do it efficiently, there's no need for computers.

The interviewer then asked, "Would you say *efficiency* is the driving force motivating computer science?" to which Billard responded,

> Yes, definitely! This started the whole field. However, now there are many applications for computers, some of which are concerned with efficiency and others which are concerned with more sophisticated capabilities. That is, you can do more things.

[6]For readers who may be unfamiliar with the term *algorithm*, it is defined by *Webster's New World Dictionary of Computer Terms* as "a prescribed set of well-defined, unambiguous rules or processes for the solution of a problem in a finite number of steps. Commonly used as integral parts of computer programs (p. 7)."

When the interviewer asked about the specific areas of work that computer scientists specialize in, Billard answered that the field primarily groups itself into *Societies* sponsored by IEEE[7] or *Special Interest Groups* (SIGs) sponsored by the ACM.[8] The IEEE has 37 individual societies and the ACM has 10 SIGS, all of which concentrate on some aspect of computer software, hardware, or systems engineering. Many computer scientists, however, have research interests that fit into several societies or SIGs and, thus, pay the membership fees, subscribe to the journals, and attend the conferences of more than one professional group. These groups are listed below. In addition, professional groups and publications that relate to Billard's area of specialization have been put in bold face print.

Societies of the IEEE

The IEEE has a broader membership than just computer scientists. It includes engineers involved in all aspects of electrical and electronic research and application. Those groups which are most attractive to computer scientists are marked with an asterisk (*).

Aerospace & Electronic Systems

Antennas & Propagation

Broadcast Technology

Circuits & Systems*

Communications*

Components, Hybrids, & Manufacturing Technology

Computer*

[7]The Institute of Electrical and Electronics Engineers, Inc.
[8]The Association for Computing Machinery

Consumer Electronics

Control Systems*

Dielectrics & Electrical Insulation

Education*

Electromagnetic Compatibility

Electron Devices

Engineering Management

Engineering in Medicine & Biology

Geoscience & Remote Sensing

Industrial Electronics

Industry Applications*

Information Theory*

Instrumentation & Measurement

Lasers & Electro-Optics

Magnetics

Microwave Theory and Techniques

Neural Networks Council*

Nuclear & Plasma Science

Oceanic Engineering

Power Electronics

Power Engineering

Professional Communication

Reliability

Robotics & Automation*

Signal Processing

Social Implications of Technology

Solid-State Circuits Council

Systems, Man & Cybernetics*

Ultrasonics, Ferroelectronics, & Frequency Control

Vehicular Technology

Special Interest Groups of the ACM

The Association for Computing Machinery groups computer scientists according to several areas of specialization based on its SIGs and/or its *Transactions*, the major refereed journals for each area of specialization. The ACM divisions of research specialization are the following.

ACM Transactions

- **Transactions on Computer Systems**
- *Transactions on Database Systems*
- *Transactions on Graphics*
- *Transactions on Information Systems*
- *Transactions on Mathematical Software*
- **Transactions on Modeling and Computer Simulation**
- **Transactions on Networking**
- *Transactions on Programming Languages and Systems*
- *Transactions on Software Engineering*
- **Transactions on Operating Systems**

ACM Special Interest Groups

- SIG3C (Computing in Community Colleges)
- SIGACT (Algorithms & Computing Theory)

- SIGADA (Ada)
- SIGAPL (APL)
- SIGAPP (Applied Computing)
- SIGARCH (Computer Architecture)
- **SIGART (Artificial Intelligence)**
- SIGBIO (Biomedical Computing)
- SIGBIT (Business Information Technology)
- SIGCAPH (Computers & the Physically Handicapped)
- SIGCAS (Computers & Society)
- SIGCHI (Computer-Human Interface)
- SIGCOMM (Data Communication)
- SIGCPR (Computer Personal Research)
- **SIGCSE (Computer Science Education)**
- **SIGCUE (Computer Uses in Education)**
- SIGDA (Design Automation)
- SIGDOC (Documentation)
- SIGFORTH (Forth)
- SIGGRAPH and SIGGRAPH LITE (Computer Graphics)
- SIGICE (Independent Computer Environments)
- SIGIR (Information Retrieval)
- SIGLINK (Hyp/Hypermedia)
- SIGMETRICS (Measures & Evaluations)
- SIGMICRO (Microprogramming)
- SIGMOD (Management of Data)
- SIGMM (Multimedia)
- SIGNUM (Numerical Mathematics)

- SIGOIS (Office Information Systems)
- **SIGOPS (Operating Systems)**
- SIGPLAN (Programming Languages)
- FORTRAN FORUM (Fortran)
- LISP POINTERS (Lisp)
- OOPS MESSENGER (mistakes in computing)
- SIGSAC (Security, Audit, & Control)
- SIGSAM (Symbol & Algorithmic Manipulation)
- **SIGSIM (Simulation)**
- SIGSOFT (Software Engineering)
- SIGUCCS (University & College Computer Services)

Billard also added that since no official organizational tree existed, to his knowledge, showing how the field of computer science might be divided into sub-fields, he thought that the University of Aizu's 28 research labs might also provide a fairly good perspective of computer science research concentrations.

University of Aizu Software Research Labs

Foundation of Computer Science Laboratory

Mathematical Foundations of Computer Science Laboratory

Language Processing Systems Laboratory

Distributed Parallel Processing Laboratory

Operating Systems Laboratory (Director: Billard)

Computer Networks Laboratory

Performance Evaluation Laboratory

Database Systems Laboratory

Information Systems Laboratory

Software Engineering Laboratory

Multimedia Systems Laboratory

Human Interface Laboratory

Shape Modeling Laboratory

Image Processing Laboratory

Computer Science and Engineering

University of Aizu Hardware Research Labs

Computer Architecture Laboratory

Computer Solid State Physics Laboratory

Computer Devices Laboratory

Computer Logical Design Laboratory

Computer Communications Laboratory

Multimedia Devices Laboratory

Computer Education Laboratory

Computer Industry Laboratory

Discussion of Billard's Specialization

In addition to getting a glimpse of the entire computer science discipline, the researcher was also interested in learning about the specific specialization of the subject in as much detail as possible to better understand how writing fit into the subject's professional activities. When asked to describe his area of specialization, Billard said,

> My general area of specialization is distributed systems...which are a collection

of processors that have access to shared resources. Distributed systems are a mechanism of hooking different computers together to share resources. And if you do that, you can improve efficiency. We are very concerned about efficiency and also reliability because you've built in redundancy with these various components, and also, we are concerned with data availability. You can make data available locally to who needs it. So that's distributed systems. They are like networks. Distributed systems are really built on top of networks. Distributed systems are the operating systems that manage them and make the best use out of them. In particular, I am interested in systems which are managed by agents. Instead of applying simple or coded algorithms, I apply more AIish types of techniques. So this is closely related to distributed AI[9].

When asked what he meant by *agents*, Billard replied,

> Agents are written as computer code. But the code usually doesn't look like operating system code. It doesn't look like an input-output oriented algorithmic process. They are not human agents, but they are computer code that has AI elements such as learning and adaptation. They are responsible for managing the resources and using the resources and making decisions about them. Another name we apply to them is 'distributed decision-makers.' So these agents are distributed decision-makers.

Billard stated that this research was related to *queuing systems*. When the interviewer asked the subject to explain *queuing systems*, Billard said,

> Queuing systems is a well-researched area. The systems have an input job stream, in our case, computer jobs but queuing systems can be applied to people waiting

[9] Artificial Intelligence

in lines, or factory conveyor belts. Processors work on the jobs until they are finished. My particular interest is in modeling two concepts: One is dynamic group formation and the other is delayed communication. Both can be studied in the context of queuing systems. However, I approach them from a completely different perspective. In my dissertation, I studied these two areas in the context of game theory, and the idea of using game theory in operating systems is just somewhat strange. So what I've done since graduate school is to go back to queuing systems and have done more standard experiments...still using the same concepts of delayed communication and dynamic group formation. Now, using models of queuing systems, I can present this research to more people and they will be comfortable with it. But I still look at the things that I'm interested in in a queuing system context.

The interviewer then asked what information the subject was really after in his research. Billard responded,

In some of my studies, I am actually extending something that somebody else did. I take the models they developed and I do further work on them or I redefine the model...it has to be something interesting and meaningful or there is no reason to publish it. And so I take it one step further. At that point, I'm not sure what I want someone else to do with it. Maybe they would take it farther. I've opened up someone else's box and added some more ribbons and interesting things to it.

Then the alternative is when I actually have some more original ideas. And these ideas have been initiated by me,...much of the theme of my work is that there are some general principles that apply in distributed systems and we often miss those general principles, especially computer scientists. You know we are not physicists. So physicists are really worried about fundamental laws and equations. In computers we are doers, we build computer systems, we build software, we analyze systems. We often deal with something that's hands-on and people are often very comfortable with that, but you miss the fact that there are some principles. Actually I approach this from my physics background and I try to identify those principles...to show that you can actually identify them and that they are appropriate in distributed systems in a variety of places, maybe even outside of distributed systems. That's why I use game theory. It captures interesting behavior. And so I'm trying to establish some of these principles. I often make models that are more abstract than people are comfortable with. People want to take a computer and plug it in and turn it on and see how it works. My models are a little more abstract than the interconnection of these computers that I'm trying to model. But because I make it at a higher level it is easier to abstract some of these principles. And these principles deal with dynamic group formation, how computer agents in distributed systems might interact with each other. And how they interact when the information they exchange is, by definition, delayed because it is in a distributed system. I'm trying to identify those principles that exist at a fairly high level of abstraction and to convince this community that it is still relevant even though they usually like hands-on activities. The existing systems may be more complicated, but you can't always get your principles out of very complicated models. That's why physics is so successful, it makes simple models and abstracts laws from them that actually work.

Queuing systems has a long history of research. There is a lot of work in it and a lot of people in it, but usually it tends to be more mundane. And that's not a criticism when I say mundane, you know...they are trying to simulate an existing system. I use queuing systems to simulate systems that don't exist now but may very soon, especially with the growth of a world-wide network. It is not the queuing system itself that I am interested in. I'm interested in higher level principles and I use the queuing system as just the playground where these principles can exercise.

When asked about how his area of specialization fit into the field of computer science, Billard remarked,

> In computer science, distributed systems is an identified research area. There are journals in it, there are laboratories in it, conferences in it. And the one phrase that describes my specialization is 'distributed decision-making under delayed communication.'
>
> There is a group of people who have...and I try to find every instance I can of publications of these people...that have published papers that deal with this area. But the area itself doesn't have that name. There really isn't something called 'decision-making under delayed communication.' I can use that phrase and people will probably understand it, at least those people in my area. But that has not been a clearly identified subtopic. And I can point to people who are doing work in the area, people like Huberman and Hogg who have a model that's been published in a variety of places that captures, theoretically, the behavior of these really large systems. The systems have a large number of agents, all deciding how to share resources, and they have old information. What's going to happen? You can describe mathematically the behavior of these systems. That's what I try to do also. I try to design models that are similar to that or even use theirs and take it someplace else. So these people exist and I wouldn't say it's a large number,...and it's not readily identified by other people, but, but...we know it when we see it.

The interviewer then asked, "Of the total computer science community, what percent would you say constitutes your specific area of specialization?" Billard's answer was

> So small that you can't measure it...less than 1%. But if you took AI, you'd have a pretty big percent. If you took distributed systems, you'd have a pretty big percent. I don't know what it is, but it would be reasonable...something like 10%. So you'd have a fair number. If you're looking at how many of those areas,..there are probably 15...maybe 10 to 20 of these major research areas in computer science where AI just happens to be one of them. And distributed systems is one.
>
> But once you break it down into any one of those, you're going to have a problem. I mean, I went to a conference in which most of these people did Bayesian networks. And everyone at that conference knew what a Bayesian network was, and they all know each other's work. But that's one conference, and they had maybe 70 or 80 people there. But that's it for the world. And then there are people who do genetic algorithms ...and genetic algorithms are very popular...so there's one conference a year and people who do genetic algorithms go there and maybe you'll have 300 to 400 people. But that's it for the world. And if you think about how many computer scientists there are, 300 or 400 is not going to be much. But it's a pretty good community if you're all looking at genetic algorithms.

Billard's Education and Personal Writing Practices

The following extended dialog took place during one of the interview sessions. Here, Billard discusses his writing education, his writing in the computer industry, and his writing in academia as a professional computer scientist. The interview is reported here at length.

Interviewer: Tell me about your writing background. What kind of courses or writing experience did you have in school?

Billard: I had 5 semesters of literature, taught by English teachers in a special department of the engineering college. The first semester was the Greek tragedies, the second semester Chaucer and that period. The third semester would be Pope,...and then the Romantics. The fourth semester would be modern literature such as *Lady Chatterley's Lover*, and things like that. The fifth semester was an elective and we covered a broad area, starting with the Iliad. In fact they were actually some of the best courses I ever had. The teachers were all very good. We had to read a lot of material, discuss it in class, and write papers about it. I had come from a high school where I had considered being an English teacher because I had such a good English teacher when I was a senior. When I started in engineering college, those were actually my favorite courses and I spent a lot of effort on my writing and it actually improved quite a bit. I would often go in and talk to my English professors about my writing. Then what happened is that I didn't really write for over fifteen years. I worked. Well, I wrote computer programs and documentation for these computer programs, but that's very...you know...you want documentation to be as simple as possible. Usually input and output...very simple. When I went back to get a Ph.D., I started writing. Before this, I got a Master's in between jobs, but I didn't have to write a thesis. I did, essentially, a review of literature and had to stand up before a committee and talk the significance of the research. It was only when I went back to get my Ph.D. much later that I actually had to start writing. Coincidentally, my professor really cared about quality writing...He certainly knows his research areas and he wants to get good research out, and all that, but we spent a significant amount of time on my writing, over and over again.

We rarely talked about results because he could understand results right away. He could understand their significance, but he really cared about the presentation of those results...and so that really put me through a several-year process of just being grilled. You'd think he was an English teacher in terms of...'What does a sentence mean?' And so he actually broke me of several bad habits I had. Now my writing is a lot more free-flowing, more fluid, I think. Whereas in the beginning, I thought ...Oh, I'm writing technical material, so therefore, it has to be dry...very simple sentence structure...and so, I got more creative again after breaking myself of those habits.

Interviewer: So as an undergrad you had writing in the engineering school?

Billard: We always had to write papers..it was a four year program, though mine was five since I got another bachelor's in business at the same time as my engineering physics degree. We always had to write papers in the five semesters I took of literature, basically that was your grade. And I think those courses were better than what the general liberal arts students got. I think they were very directed and focused and the teachers were very good.

Interviewer: So you were actually not writing about engineering.

Billard: No, there was no technical writing course in the undergraduate engineering curriculum...I have never had a technical writing course. I think that was a mistake. It wasn't in the curriculum. It wasn't in the engineering curriculum, I'm sure it was someplace else, but it certainly was never made available to us.

Interviewer: So most of your writing was expository.

Billard: Of literature...I didn't really have to write technical papers as an undergraduate...I was an engineering physics major, so you do a lot of experimentations and you write up experimental reports...you learn the scientific method very early on with that. You learn how to state hypotheses, and the experiments and the results and the conclusions. We knew the organization, we were taught that very early and we always carried that throughout, but there was never really any exposition that went along with that...it was always very simple.

Interviewer: Did you ever have any writing instruction or conversations about writing with the engineering faculty?

Billard: No, because as undergrads you just don't get the exposure. You don't get to see them and meet them. Of course at this university (University of Aizu) you have the opposite. I did have some very good physics instructors. These were not engineers, so although my major was engineering physics, I got all my physics courses from physicists. They were all very good...almost all of them were British. They were good teachers and I had some close relationships with them. But again, that was not about writing, it was just about physics. And I was not really doing any real research, so there was no...well, I did write some papers. I wrote some papers for my physics courses, usually reviewing something, but there was never really much critiquing of the writing. There wasn't any, really. That wasn't the emphasis.

Interviewer: Did you ever publish anything as an undergraduate?

Billard: No. Well, I had two summer jobs where I worked for physicists and so I would run experiments with them. One summer...it was at a nuclear physics lab...I would run experiments with the professors, but I didn't participate in the writing of those results. And another summer I worked closer with a professor and my name got put in not as an author, but as an acknowledgement. But I did not write any of the results of those.

Interviewer: And then you worked as a computer programmer?

Billard: I worked as a programmer for several years and then I went back and got a master's degree. I continued to work part time in teaching and software engineering. And then I worked about eight years full time as a software engineer and software manager building database applications.

Interviewer: Before you went back for your Ph.D., during your term as a professional programmer, did you ever have to do any writing?

Billard: I wrote documentation of two types: on-line help files and more formal manuals about specifications...well sometimes they were specifications of things that were being proposed. And those needed to convince other people that what we were going to build would satisfy the needs of the customer. And those were fairly important and difficult to write. And then there were on-line help files and some more programmer-oriented documentation about the specification of how to use these systems...It wasn't scientific. It wasn't really highly technical.

Interviewer: Did you have any models you used to imitate other documentation?

Billard: Unfortunately not. Most of the documentation was kind of free ranging. That was probably a bad thing. We developed some models later on where the style of the manual conformed to other things that I had seen, for example the UNIX system's on-line help files. But I didn't really look for other examples. I just sort of thought these things up myself. Which has some element of creativity, but it may not be the best thing.

Interviewer: Now when you went to grad school was there any kind of writing instruction at all?

Billard: None...well the first couple years you just take courses, and you would have to write for some of those courses,...it was really only when I started working with my advisor that he really started getting on my case about my writing.

Interviewer: Would these be technical reports that you'd be writing, or...

Billard: Umm..., they were more open-ended project assignments where you had to think of something original, do it and then report on it. But the structure of the results usually was very free-form. You didn't really have to follow any heavy duty technical layout. So there were pretty much open discussions about what you did and the results.

Interviewer: And then you started doing research together with your advisor...

Billard: Yes, that's right.

Interviewer: And then you started writing together, or...

Billard: No, I was creative enough that I could come up with my own ideas. So, I didn't need a lot of direction from him. I was older and had my own scientific capabilities. Other students might need a little more push in terms of ideas and what they needed to look for, but I already had a set of ideas that I wanted to look at. We had a meeting every week where I would report my progress on particular projects—all of my advisor's students did this. And I would put together a slide show every week. I would try to convey as quickly and effectively as possible what my ideas were. I would carry this on through my research and then when I got to a point where the research was yielding results, then I would actually write. I would write this at my own instigation. I wouldn't ask him 'Should I start writing?' I would say I am at a spot where I can actually report on this and what I did was write a paper. The paper might be anywhere from ten pages to twenty pages long about this set of results, basically. Then he would read them...and he would be busy...but he would read them...and mostly the response would be on the exposition...some of it on the results. I wrote various reports, and when I wrote them I had them in mind as being chapters in my dissertation. Then those chapters went off to different publications, either conferences or journals.

Interviewer: So if you were to identify different genres or different kinds of writing that you did in your Ph.D. program, there would be something similar to a technical report?

Billard: Yes,...but there is something that goes on before that that is the milieu of science that you do before the technical report. That's where all of the scientific effort really takes place. I call that a working paper. That working paper I go over twenty...thirty times. Just myself, in terms of what's in it and the phrasing and presentation. I spend a lot of time on that working paper. And once I'm done with that working paper, there is not much to make me change things unless somebody comes back and says "I don't like it." Because once I've got to that stage...I've got my results in there...for the rest. of the publication series the results don't change. But the presentation does change based on the targeted audience or how a particular set of reviewers reviewed it. A different set of reviewers...they'd have a different set of comments and the paper might change in different directions. But most of that is usually oriented towards convincing and helping the reader (reviewer). When I write a paper and go over it twenty times, I understand it perfectly. And you'd think that somebody else would, but of course they don't because this is not their exact area, they didn't do your research. And so, every time you have a fresh reviewer, they're a blank slate and they're just kind of reading it and absorbing it. And places where they have problems then come up in the reviews and then I try to modify that area. And that is really the only thing that makes me change things, that external force. Once in a while I realize that I just haven't given a novice reader enough pictures to convey what's really going on, and so, sometimes late in this sequence I will actually develop some more pictures. This often occurs as a result of making a slide show presentation. If I'm in a conference I have to make a slide show presentation, and then I have to think of a whole different way of presenting the information. I may have fifteen, twenty, twenty-five minutes to present this, and you need some really simple, clear pictures...and sometimes I say, "Oh, wait a minute, this picture is really a good way of presenting the problem or information" and actually it would actually fit in the paper. If I haven't already sent the paper off, I can take some of these new pictures and get them into the final published version of the conference paper.

Interviewer: So then as a graduate student you wrote working papers which are somewhat equal to a technical report...

Billard: I could have published some of them as technical reports at the University of California, but I just didn't do that. I didn't have a real motivation for doing that so they stayed as working papers and what I actually did was build a structure around them using LaTeX as my word processor to build my dissertation. So my dissertation is not one monolithic piece of text. It actually has hundreds of pieces of text that can be regrouped as chapters or regrouped as conference papers or regrouped as journal papers. And that was hard to manage but the alternative was to be writing the same thing ten times over and trying to keep them all consistent. They developed as working papers, and I have a hard copy of my dissertation, but the text behind it still consists of hundreds of files and diagrams and results.

Interviewer: The research that you were working on at the university, was it all related to one particular project, or...

Billard: One project. You have to have one real strong theme to your research because that is what they are going to pound away at, both your advisor and your committee. You have to go through these various stages of acceptances. And they have to make sure you're really focused on some key things.

Interviewer: And that was with queuing systems?

Billard: Well, actually this was not queuing systems. My advisor works in queuing systems and his dissertation was on queuing systems and so his work was actually a starting point for mine. And his previous Ph.D. student was actually doing queuing systems, and because of that I said 'I don't want to do exactly queuing systems'. I was interested in the two concepts: One is dynamic group formation and the other is delayed communication. Both of those have a lot of applicability and can be done in queuing systems. That's what a lot of my research is now... looking at these topics in the context of queuing systems...but then I didn't want to have my dissertation look too much like the guy who just preceded me, or like my advisor's–his Ph.D. was only a few years earlier–and so I actually stayed away from queuing systems.

I did something that was fun and enjoyable, and I actually got a lot out of it, but it was difficult to publish, and ...well, now I'm finally getting most of it published, but it was more difficult because I did it from a completely different perspective. I did it using game theory, and the idea of using game theory in operating systems is just somewhat strange. What I've done since then is gone back to queuing systems and have done more standard experiments...still using the same concepts of delayed communication and dynamic group formation.

Interviewer: So at the university level, you did these working papers that all fit together to eventually make your dissertation. Any conference proceedings?

Billard: They came out of that. I never wrote a stand-alone conference or journal paper at the university with my advisor. All of them were in the context of my dissertation...the papers flowed out of that. And that may be atypical. Some people might work with an advisor on some of HIS research and get their names as a second author on HIS research ...to get their foot in the door. But I was already off and running with my own research with my advisor letting me get on with it.

Interviewer: Who do you write for? Tell me about the audience you write for in your papers?

Billard: I write for the general population of computer scientists in my field. I might know a few people that I would hope would read my papers. My work references their work...some of my work is very closely related to some other people's work and so, I hope that they would see it. But that is not really who I am publishing for. These people who I am publishing for, I might know some characteristics about them...but there are two levels. One answer is 'Whoever buys the book, Whoever goes to the conference and gets the conference proceedings, Whoever subscribes to the journal' is a potential member of the audience. But usually that audience is much broader than that–though most are computer scientists– who actually reads it. In referee reports, one question they ask us is 'What percent of our audience would be interested in it?' It isn't supposed to be 100% because that won't ever happen. Is it 10% or 20%? This is actually a very good percentage if they open up this journal and say 'Oh, I want to read this.' That number is going to be significantly smaller even though the people who subscribe to the journal may be in a fairly tight specialization.

As an example, I got a journal yesterday on parallel and distributed systems. My area of research is distributed systems. It's not parallel systems, so unfortunately, that journal is double the size it needs to be, and they usually emphasize parallel. And so, because of that, I usually don't read most of the articles in there. But when I got this journal yesterday, I was looking thru it and 'bam' there's one that stands right out. It's exactly in my area. So, you know, I drop everything and I read that. It's only because that one particular paper is so closely related to what I do that I had to see what those results were.

The audience in the general sense are the people that subscribe to that journal or that go to that conference. But even conferences like AI (artificial intelligence) or in distributed systems, if you look at the call for papers in these they are extensive, in terms of the number of topics they have. Although it's called distributed systems which is one area of specialization in computer science there happens to be all these topics, perhaps 20 or 30 or 40 topics. And so, only some people will come to my talk. You know there are multiple talks running concurrently, and only some will come to mine. And, usually, you're very happy if you have one or two people in there who know exactly what you're talking about. Because you can tell by their questions afterwards that they know exactly what you did and they have some pretty tough questions sometimes. And so the audience comes down to a small handful. If you really want the people who know exactly what you are doing...These people will usually be computer scientists and they are usually in my area of specialization, distributed systems...something to do with that. My research does branch out into areas like game theory, and in that case it would have a different audience that I can really identify when I write a paper like that.

I have never written a paper for the general computer science audience. I could. I have some interesting results and I could phrase it such that 'Here are these results that you could understand in some general context.' But I haven't done that because...maybe my work is not mature enough or it's not well known enough. Most of my work is very targeted for those who are in my area of specialization and can understand what I am doing.

Interviewer: What do you hope to accomplish with your writing?

Billard: Well, the research I do doesn't exist unless it is published...I mean, I could think away all day and do experiments all day, but without the act of publishing,...Well, my work doesn't exist unless I write.

The main function of my writing is to communicate the results of some scientific experiments. So that is what I'm doing...that is the purpose for almost entirely all of my work here with two exceptions: software and education. We develop some software here. Partly, I need it for my research and partly, I need it for my students. Because we try to do interesting software development there are places to publish. Now there are people who do research in software, but I don't. I'm not a researcher in software but I still build software. And because I do, I want to publish it. I publish in different places and with different presentations because I'm not presenting it as a research result. I'm presenting it as interesting software and software design. Now that's out of my main topic but I think people will be interested. Another small activity, separate from my research, is writing for educational publications such as *IEEE Transactions on Education.* We have designed software and laboratories for our students to use and I'm trying to publish these activities. Again, that is a small part of my effort...Where I publish and how I publish and what I write looks different from my technical work. I have three purposes. One is to present technical results, the analysis of very detailed technical results, to an audience that is very closely related to that same thing, or to present the results of developed software to the software community in general, or to present the results of educational tools and associated laboratories to teachers. There are three activities and the three audiences.

Interviewer: So who are your peers or colleagues in your area of interest?

Billard: The people that I'm really trying to convince are the people in distributed systems. And they tend to be oriented just toward the networking aspect of it. But the principles that I am trying to apply really look more like distributed AI in which the agents have a higher level of abstraction of what's going on about them. We are not concerned with protocols...we are concerned with treating these resources as objects and trying to successfully put these objects together to improve performance or achieve some other set of goals. And so I've actually published some in AI formats because they tend to be able to very quickly and very comfortably understand what I'm saying because they're used to taking these high level abstractions and talking about distributed decision-making and agents and adaptation and learning. But I don't pretend to be an AI person. I've absorbed some of that from the work that I did with my advisor. I'm still trying to convince the other audience that is not so comfortable hearing about distributed decision-makers.

Interviewer: So, you're writing for both people within your specific area and people kind of on the margins of your area. Is that right?

Billard: Yes, but you see...the number of people in my specific area is so small that you don't need a journal to do that. And so you really need to convince people outside of that group. That's the harder part because my work doesn't fit into nice pigeon holes, especially the way I go about it because...maybe because of my own personality or because I came to this late...I had other ideas, and so I went about it in my own way, rather than being born into the field. I needed something that was at a higher level and dealt more with principles. And that's more difficult. And even if you come up with them, it's harder to convince people of their validity and their usefulness. Especially computer scientists like to see, 'How can I use this?' Where a physicist will see the value of having the principle itself. But the computer scientists need to apply it. And that's not a criticism, I mean that's why computers are around here. They are here to be applied.

Interviewer: Then, in your area of specialization, what would be the main conferences that you would try to attend to disseminate your knowledge? What conferences do you aim at?

Billard: There are three major conferences in my area that are recognized as the top ones–Principles of Distributed Computing (PODC), International Conference on Distributed Computing Systems (ICDCS), and Symposium on Parallel and Distributed Systems (SPDP). They are all recognized as the top-flight distributed systems conferences...two are IEEE and one is ACM sponsored. These are recognized as state-of-the-art, top research, with the most interesting results, most applicable to the largest community possible.

Interviewer: How many papers do they accept at these?

Billard: Well, PODC would be a very small number, say 40 and ICDC maybe more like 80 or 100,...it isn't a large number. And so, there are also journals that are thought of as top journals.

Interviewer: And, what are these journals?

Billard: *IEEE Transactions on Computers* is the best one for disseminating to the general computer science community. To get your results into there establishes your work, even though we all have our specialties. It can now be presented to a very large number of readers ...it is significant work that they can still understand. Some more specific ones are *Parallel and Distributed Processing*, and *Software Engineering.*

Software Engineering is a fairly popular journal...and years ago they published things more related to operating systems and operating systems' performance; they thought, well, to solve those things you need software engineering. Most of those things are not published in there anymore. So, even though most of my work references things that were in that journal, they won't publish my work. They have made a conscious decision to move that topic out of the journal and it's gone to other places. And so, my work would not really fit in there anymore. *Systems, Man, and Cybernetics* is where I get published,..I've had four accepted there, and I've submitted another one. And it is because this journal, *Systems, Man, and Cybernetics*, well...it about covers everything. People that review and read it are much more open to complicated systems, that is, higher level abstractions to deal with these very complicated environments. That's why I feel confident...comfortable submitting there.

Interviewer: What publications would ideally represent your field?

Billard: Let's go back to this journal,*Parallel and Distributed Processing*, this is a journal that I picked up yesterday, and I don't read most of the papers in there because most of the papers are on parallel environments. But I looked and there was one on distributed systems and it's very closely related to my work. All of their graphs,..their X-axis,..basically it is the X-axis that I care about...how much delay is built into their system. They are modeling distributed databases. I don't deal with databases, but the fact that these were databases didn't matter, even to these guys. Although they had a real system they were going to simulate, the fact that it was a database didn't matter. The important fact was that they had multiple processors processing database transactions, and these database transactions looked like jobs. And that's what I deal with. I deal with processing jobs. And they were looking at the same problems, "What kind of algorithms can we come up with that will make this system efficient?" given that the processors or agents or decision-makers have old information about the loads on the processors. This is closely related to what I do,...identical to my research effort. But I would have a difficult time getting in here.

The reason they got in here is that they did a great study. It's really top-notch. But they also made a very complex study. And you can get in trouble doing that, but they did it successfully. And only because it was complex was it convincing. The editors and reviewers for this journal still want to see...you, know...it's really got to feel like a real computer system before they are convinced. The researchers had a very complex model...I would never attempt to make a model that complex. It is very easy to make it complex, but if you make it complex, no one will understand it and they'll say "Reject." So, you have to be very careful. These guys made it complex and were still convincing,...I mean I was convinced and I'm sure the reviewers were convinced...that it was still reasonable and understandable and relevant. But when they were done, they said, "Well, guess what! You know, our model is so complex there is no way we can possibly do a simulation of it because the possible alternatives for all of these parameters that we've set up would be overwhelming." And so, they had to admit that in their paper, and then say, "Well, because of our experience we're used to dealing with these systems, and so, we know basically how to set this one parameter. We don't have to try every different possible value for this parameter. They did a great study, a lot of effort, very complex models, they pulled it off; they convinced these people and could publish it.

Almost all those categories, I would fail at. Because I'm not senior, I don't have a lot of research history. If I built a complex model...it's very difficult to do that and be convincing that it's just not complex for complexity's sake. And so I tend to build simple models. Because of the way that I am going about it, if I made it complex, I would lose. I try to build simple ones to extract the same principles or even more principles than what these people showed. I actually can make even stronger statements about when you're going to have problems than what these people can say. But what these people showed was a very specific instance of this at work. And because they were willing to do that effort and they wanted to get their hands dirty, they were successful. And it is a great paper.

As for my papers, I've seen the reviews. If it doesn't really smell like an operating system, a real system they know about, the reviewers are not convinced. Most people deal with real systems that are too complex to analyze mathematically. That's why they have to do simulation. Simulation means running experiments using a simulator. But if you want to do analysis on these systems to actually mathematically predict the behavior, you need a simple model. And it tends to be more abstract. And then it tends to be less convincing to these people that its relevant. That's where the problem is.

Interviewer: In general, do computer scientists first target a particular journal or conference and then do research that fit the goals and standards of that journal so they can get published, or do they see some need, do interesting research in response to that need, and then ask themselves, "Now, where can I get this published?"

Billard: Both of those could happen, but it is probably the first one that is the most common because of the evolution of the species. Those that do that succeed. When I started, I didn't do that because one, I didn't have any experience in publishing, two, my advisor didn't have a long publication history, and three, I was older and had my own ideas. I had several years of open thinking, looking at other ideas, but thinking about "What else is out there?" I came up with the studies without knowing where to target them. And then we really had a hard time to publish all these studies. "Where are we going to publish these?" Of course, my advisor immediately thought of top-notch conferences and journals. He liked my work but the real reason was that those forums are the first to come to mind. Everybody knows them. And it was only after finding other places and refining those original works, that I was more successful. Now what I do, is to target specific forums.

Interviewer: Since you've come to this university, is the research you are doing continuations of what you were doing for your Ph.D., or...

Billard: The concept is still the same; the area's still the same. What's happened is I've learned more. The act of having to do the dissertation and then seeing other papers...I broke some barriers. After understanding some concepts in detail, I finally broke through a barrier where I could redefine the model that I worked on in my dissertation...make it much cleaner and much more powerful in terms of what results I could get out of it. Now I could simulate these systems and mathematically analyze the results. I kind of hit big pay dirt, from my point of view, in terms of creativity and productivity. My ideas reached some critical mass that they all kind of fit together. It's all a continuation of my earlier work...really a strong refinement of what I did earlier. New models have come from that, but then they all have the same theme of dynamic group formation and delayed communication.

Interviewer: And the kinds of work that you are doing here are typical of the kinds of work other people are doing in distributed systems?

Billard: No,..I think I'm a maverick because I'm trying to extract some high-level principles, that are important, that exist in these systems. If you have a distributed system, you have to be concerned with group formation and delayed information. And so, I have a real strong personal feeling about this. It is not something I do because someone told me to do it or I have to do it. I have some really strong personal intuition and motivation to actually look at this area.

And you asked about how this knowledge is going to be used. Is it going to be sold or packaged or for industry. It can't be, because it's really too abstract. It's really meant to be some general principles that describe the behaviors of these systems. And hopefully, from that there may be some practical use as we build larger and more complex versions of these systems. We don't have to ask for that to be done, it's already happening. We can't stop it. And without understanding these principles of behaviors with large numbers of interacting agents in highly distributed and complex systems, you can run into trouble later down the road. Or, we might miss opportunities that we can't take advantage of because we didn't even think of them without, perhaps, these principles. So that's my guiding light.

I had someone come here who is a consultant for MITI (Ministry of International Trade and Industry). They wanted to talk about how research could be done and be made available to local industry. But because of my situation here, since I am on a three-year appointment, and because I'm a new faculty member, I don't have a large window in terms of publishing. It takes a fair amount of effort to do it and actually get it published. And so, I don't really have a long window here.

I am a very practical person, and I have done some very practical things in computer science. I built real systems that sold for a substantial amount of money and are used in very complex situations. But they are very time-consuming to implement. And if you want to do something like that, it's going to take you five years. And actually in terms of publications, that's often what they want to see. They want to see real systems. Because it has to be relevant. And if it appears too abstract, then it appears meaningless...even though the abstractions and the principles in the end are going to be important. If you're going to do something with industry and actually use it, it's going to take five years. Five years to work with them, to figure out what the problems are and develop solutions to solve them and publish them. All that is a minimum of five years.

Interviewer: If you were approved and tenured here after the third year, would you pursue a more practical course, or...

Billard: No matter what happens, I still have a strong belief in what I am doing, but I think I've also reached a plateau,...I don't know whether it is the top of the mountain, or if it is just a plateau, but I have really been able to express in my writing the things that I was looking for. And I had to do the research before doing the writing. But I've really come up with the models and the kind of results that I have had inside of me. I didn't know it, but they were inside of me...somehow. I've been able to come up with those and I'm satisfied with the results. So that is where most of my effort is right now... going to different conferences and submitting to different journals and resubmitting. They come back with a certain review and sometimes I have to resubmit to the same place again. After that, I may do something practical. I will probably do something that's related to industry. It may not be in this area. I really enjoy software in general, though my area is in operating systems and I have a special research niche, I really enjoy software in general. I may step back and do something else, and I don't need a large number of publications to come out of it, I don't need them quickly.

Back to my research, there are principles that govern this area and I have a very strong belief that these principles are important. We often think that there aren't any principles. We just simulate these systems and observe their behavior and say "Oh, look! Let's plot a graph and these are the results." But there is something that is driving these results. And that's what's driving me. I believe these principles are important in a lot of other areas. That's why I'm interested in game theory. Because of the idea of dynamic group formation and making decisions when you don't have instantaneous information, you can apply that to people. There is something very universal, I think, about these principles. And so the reason I write, is first of all to convince the people that are usually more nuts and bolts oriented in terms of distributed systems, that these principles are important. And second, to try to show that these principles are important other places such as economics and biology.

There's a topic called delayed differential equations...differential equations describe changing or dynamic behavior. It's calculus but with delays built-in. What's interesting is that biology has done this for forty years. It was a biologist that developed these equations and how to solve them. And, computer scientists don't know about them. If you look at people like Huberman and Hogg, they're not really in the mainstream of computer science. They publish their results in distributed AI. And most of the people in computer science don't know about them. They don't know that there are standard mathematical techniques for solving these kinds of equations. I've learned about them, so now I'm trying to apply them in a computer science environment and show them to computer science people. Coincidentally, I've just recently got a paper accepted in a biology journal using these techniques applied to game theory. So, the same theory that the biologists started forty years ago is coming back to them from a computer scientist.

Interviewer: So your mission is to raise the consciousness or the...the level of understanding in the field itself...

Billard: Yes,...using abstractions of things that really do exist, but they are high-level abstractions. There are certain principles that we can find out about the behaviors of these systems. And one of the tools we can use is delayed differential equations which have been used by, first of all, biologists and certainly by mathematicians.

There are conferences in mathematics in delayed differential equations. But very few computer scientists know about it, and...well if you're talking about distributed systems, you ought to be concerned about delayed information. Because the information will be delayed especially in a global environment. There will be too much overhead in communication cost, and you can't always disseminate the information that is needed to make these decisions. And so, as systems get larger and more complex, these obstacles exist, and my goal is to show some techniques that have been used other places that could work here.

Interviewer: So, now you have this backlog of knowledge. And now your primary focus is on how to get this published rather than...uh, making new knowledge...

Billard: Right now.

Interviewer: Earlier in one of our interviews you described computer science. Would you classify yourself as a computer scientist?

Billard: Yes. Definitely. But, you see, computers is a very big industry. There are a lot of people in the computer field that aren't computer scientists. A computer scientist is involved in the systematic STUDY of these algorithmic processes, etc. There are many people that write software. I was a software engineer and a programmer, certainly not a scientist at that point. You have to be involved in the systematic STUDY of those algorithmic processes that describe and transform information. And you can do that with theory; you can do that with implementations of applications...So there are different ways you can be involved in that systematic study, but you still have to be involved in the systematic study of it to be a scientist. I'm definitely in that category.

Interviewer: And how important is writing to your work as a computer scientist?

Billard: Crucial! If I don't write, I don't have work...mainly because I'm not teaching a formal class here, though I do teach students that have joined my research lab. I'm going to start teaching next year, but my activities for now have really been research, and the only way I can finish the research is to write it and to publish it.

Interviewer: And what kinds of writing do you do?

Billard: There are three kinds. The major one is the reporting of technical results. And the audience for that is computer scientists in general, and more specifically those who are more interested in distributed systems. And then to a smaller degree, I write about the software that I have implemented, which is not my research, but it is important. I either needed it for my research or for my teaching. It is actually very interesting software, in the sense that it is just not code. There is something interesting about its design and the way we actually built it. And so the audience there are other software engineers who are building software. And another small category is for teachers, publishing educational tools and laboratory exercises that we are developing for the lab here. Those last two are smaller categories, and aren't really research, but still important.

Interviewer: What kind of document do you write most frequently?

Billard: I write working papers.

Interviewer: Now, what's the difference between a working paper and a technical report?

Billard: A technical report is a final working paper, the final version of the working paper.

Interviewer: So you are continually working on a text while you are doing your research? After you do your research?

Billard: No, it's after. I often have a picture in my mind about where I am headed and what the document will look like when I'm doing the research. But usually the results are all in my hands before I write anything. I have the results in my hands, and I usually have the graphs and figures. Then I start writing.

Interviewer: So the working paper would be rough drafts of the technical report?

Billard: Right.

Interviewer: And then the technical report would be the first version of your research that you would make public.

Billard: Right. And it happens when I've gone over that paper so many times, perhaps 20 times, and nothing else changes. When it stops changing, then I usually need other reviews, for example Alice[10] here or my advisor in California, and they may start reading it after I'm done.

Interviewer: So you take it as far as you can, and then when you can't go any further, your give it to Alice or Dr. Pasquale?[11]

Billard: Yes, and those two are very good at reading and seeing things that I missed. Not necessarily in the results, but in the English. We're not talking about science, but how the exposition flows and what isn't understandable and what is awkward. After I read it 15 times, I just don't see it anymore. Then I give it

[10] Alice Riedmiller is a Research Associate who works with Billard in the Operating Systems Laboratory.

[11] Joseph Pasquale is an Associate Professor in the Department of Computer Science at the University of California at San Diego. He was Billard's dissertation advisor and now works with Billard on joint research projects from time to time. They have published eight papers together.

to them and they see problems right away and give me suggestions on how to fix them.

Interviewer: So, after they've read it and you've gotten their feedback and made some additional changes, you publish that in a technical report.

Billard: That's what I have been doing here, because I've had a large number of results. I knew that it was going to take a long time to get these published, so I put them into technical reports. There are some activities that I did not put into technical reports, such as laboratory exercises. I didn't think that would be appropriate. Some people here at Aizu are doing group projects on top-down education[12], in which case the educational tools are the output of the study, and so they might be published in a technical report. For me, that's more of a sideline of my business that I'm doing right here. I didn't publish that as a technical report. I sent those directly into an education journal.

Interviewer: Now, the genre of technical reports that we have here...Is this a common genre at all universities in computer science?

Billard: Yes,..at all computer science schools. We call them technical reports. Other disciplines must have something equivalent to this, but we always call them technical reports.

Interviewer: Now, after you create a technical report, some stop there and others go on to other forms...?

Billard: Yes, but some skip that. Some that are not really technically oriented,...the software designs or the educational tools, I'm not going to put those

[12]courseware development projects currently being funded at the University of Aizu

in technical reports. I'm just going to try to publish them in simple formats for the appropriate audience of that journal. And I will have some technical reports that will halt, that will not go on further. And they won't go on further for several reasons. One is they were important to encapsulate that information, but the information by itself is not that substantial enough or interesting enough to publish elsewhere.

Interviewer: Now for those that halt, were you primarily writing those for yourself to clarify in your own mind what you were doing, or were you thinking of a specific handful of people that you wanted to communicate these ideas to, or...

Billard: Yes. If someone asks me for work that I'm doing, I can hand them a technical report. And sometimes, some research work is a little bit more of a dead end. It still has some importance, but it just kind of terminates. It's scope is not as big, it's not as applicable...the results are real, in a sense, but they are just not as strong. And so, several of my technical reports, probably, will just stop there. In some cases, what flowed off of the original work turned out to be more interesting than what was in the first technical report. And so, I wanted to make sure that I encapsulated that core part of the tree. But what came after that was much more interesting and much stronger and clearer, and those became technical reports on their own, and then those will go on into, hopefully, other publications.

Interviewer: Now, do you put all of your technical reports on the network? I think we were given the option here to.

Billard: Yes, you want to put all of your technical reports on the network[13].

Interviewer: But once you've put them on the net, you've essentially disseminated

[13]connected to the University of Aizu's World Wide Web page

them, probably more broadly than a journal, right? It's just that they don't have the status.

Billard: They don't have the status, they haven't gone through a review process. But it's also on an on-call basis, so someone has to come looking for it. And they're not going to come looking for it, but they may open that journal and say "Oh, this is it!." Or they go to a conference and you give a talk and they are interested by your talk.

Interviewer: Those that dead ended?

Billard: Those that dead ended, they had a boundary around them. They weren't just some arbitrary results. There was a model; there was some simulations or analyses and there were some results. We can call those dead ends. That's not derogatory here. Now of the 13 that I did last year, most (10) will go on to other publications usually journal articles unless there is some other forum that becomes available.

If I see a conference call for papers, for example, and I think this paper has some interesting results that could be applicable there, I might extract them from the technical report and go from there. But at the moment, I think 3 out of the 13 are basically at a terminal state.

One of them is *Autonomous Agents Sharing Global Resources without Communication.* It was the first paper I did when I came here, and it was targeted for a particular place, a conference on autonomous decentralized systems. And what happened was, there was another paper that started after that that used one of the results in there. It was one algorithm, and I took that and I applied it in a whole different context, and that paper was accepted at the conference. And so it played its role; it laid all the foundation, that one algorithm,...I then pulled away from there and...it was used as one part of a much bigger study that went into a conference paper. So, they all have their roles like that.

Interviewer: Now, how about the software? Have you done all three kinds of documents (technical report, conference proceeding, journal article) here as well?

Billard: I could, but I don't want to put that effort into it because its not my main area. And I'm not trying to establish credentials in that. What I've done is submitted one small paper to *ACM Software Engineering Notes.* It's not a refereed journal but a SIG (special interest group). You send it in in camera-ready form and if they think it's interesting they just put it in. It's a really good way of disseminating information about software. People who read it who are into interesting software development. I get e-mail from these people and I supply the software. As for the educational tools, I've submitted one paper to *IEEE Transactions on Education.* It is really related to what I'm trying to do here with students; a 5-page, single-spaced, double-column report on how to teach queuing systems. What do you need to do that? What software do you need? What exercises? It's a forum for people whose primary responsibility is teaching.

Interviewer: You said that you enjoyed English classes as an undergrad. Those probably helped you with your thinking skills or organizational skills, or...

Billard: Yes,...but I lost most of it in terms of my syntax. And so it was only by getting it beaten back into me, basically, by my advisor, that I was able to improve my writing. Well, actually, I'm the kind of person who has anti-writer's block. I mean writer's block means you don't know what you want to say. But I really know exactly what I want to say. I've got it all in my head. And it's actually frustrating that I have to get it out of my head and through my typewriting into a textual format. And so, it's actually kind of a painful process because I know exactly what it is I want to say. And to get those words out and match that inside picture is REALLY difficult.

Interviewer: Do you pretty much organize the entire document in your mind first before you actually begin to write?

Billard: Well, you e-mailed me questions about specific genres in computer science and whether there are "conventions in a computer science document that you MUST conform to." In terms of the word MUST there are really two things that you're talking about. When you send something in, the editors say some MUST things, for example, it must be double-spaced, it has to have wide margins, it has to be written in English, it can't exceed so many pages,...so those are some basic rules. But in terms of what's inside the document there are also some MUST expectations.

Interviewer: So what are the musts that are unwritten but you know that your peers are going to be looking for?

Billard: I'd say an informative title and an abstract,...sometimes there are limits on the number or words that it can be...that says something about the model that we're working with and the results. And I've often missed that. I've hinted about the results because it's hard to put in a little abstract what your results are. But you really do have to say something. If I don't say something, it always comes back from your reviewer, saying, you know...you have to do that. It's something that a reviewer always looks for in the abstract.

Interviewer: So in your field, the abstracts are really mini-versions of the entire document. So a person can read the abstract and not have to read the document?

Billard: That's the idea, but it's not a mini-version in the sense that you can't really describe the model in detail. But you have to give something...either it's a model that everyone knows about and you have to say those key words so that they plug into what that model is, or you have to describe it in enough detail so that the editors can understand the domain that you're talking about.

Interviewer: So it doesn't have the purpose of whetting their appetite and trying to entice them into the article?

Billard: No,...and that's the problem I used to have. And Joe, my advisor, hated it. I used to write in this kind of "Dance of the Seven Veils" ...you know, I like to unveil...If you were the author of a novel, that's what you want to do. You wouldn't want to say on the first page what was going to happen on the last page....you want to bring the reader in and then things kind of evolve and get more clear. So that's what he had to break me of...that habit.

Also, what would happen is in writing even a paragraph, my strongest sentence would be my last sentence in the paragraph. And I would usually try to work my way down through the paragraph to the point in the paragraph where I wanted to make my point. I felt uncomfortable making a statement that I hadn't really proven yet and built up to yet. And he would start reading these and he would always go to my last sentence and say "Make that the first one." And that may not be the way to do it in other environments, but now I'm convinced that for technical writing, you need to do that because these are very difficult papers to read. There are a lot of things going on, and you need to really give signals to the reader...here's where we are at and what we are going to do.

Interviewer: So, you make an assertion up front, and then explain it and prove it.

Billard: Right. And that's more difficult to write it that way sometimes.

...Okay, so we are still on the MUST expectations. It is important to describe the context in terms of related work. One of the questions that is always going to be on the referee's report is...did the paper really describe all the important references that that referee knows, and if he knows some other references, he will often tell you. You have to state the purpose of the paper and the SIGNIFICANCE of the results. To say that the result is X=2 is not enough. You have to say why is that significant, why would anybody else care, and what does that really have to contribute to. The paper will also have a concluding section. And sometimes that might be the place to say what new work might come from this.

Let me look at some of the questions on a referee report. "Does the introduction state the purpose of the paper?" This is one of the expectations. The peers are expecting me to state the purpose of my paper. "Is the significance of the paper relative to the existing literature explained?" So, that's related work. "Is the paper clearly written and organized?" "Are there adequate references to other research?" "Is the paper cogent?" "Does the author explain well what was done?" "Does the author explain well why it was done?" So you always have to make decisions about why you did it this way, and...you have to be able to say what you did....You have to hit them over the head with a hammer which is not the way you write for English classes. Now, when I wrote for English classes, I didn't write that way, and I wasn't taught that way. But in computer science you have to be very explicit, and you sometimes have to say it more than once, but you have to be careful about how you do that in journal papers...because having said the statement once you don't need to repeat it, and there is a tendency to do that, but you have to say it one time and say it really strongly.

Interviewer: Do you think this has come about because of the vast amount of information that people have to process in computer science and they need to be able to get the information quickly, or do you think this has come about because of the large percent of non-native English writers that are publishing in the profession, or...

Billard: Computer science is really application-oriented. And so, the people that work in it tend to be hands-on, problem-solvers, and the thing is, if you look at papers that are written in physics or mathematics, they are not written like this. They are very mathematical and they expect the reader to just take the math...you know, you read it, if the math is correct, that's it. It just doesn't seem to be that way in computer science. There needs to be more exposition to really bring about a broader community that is much more practically oriented. And so in math, there is a very narrow community and it's very mathematically oriented. You can get away without exposition, in fact exposition gets in the way. In computer science, you have to give very explicit exposition. I guess you have to hit them hard because you have people coming from so many different directions, not necessarily because they're non-native English speakers,...that would certainly make a problem, I think. But it's just because computer science is so broad and it involves people that are doing a variety of things, most of which is very practical and application-oriented. And the final question on this referee report is, "Is this paper appropriate in scope for this journal?"

One other item is that we usually give a short outline at the end of the introduction section. This states what the following sections will cover.

Interviewer: So before the reader really gets into the paper, he has already had an overview of the entire paper and learned the results.

Billard: What often happened in working with Joe, and this was just part of my training process with him, was he'd really tear apart the introduction section. Usually almost all of the comments were in the introduction section. Once I got going in the model and getting into more of the math, things would be fine. But the real motivation takes place in the introduction section. " What is the environment you are talking about here?" "What is the significance?" All of this needs to happen early on. Maybe you think it shouldn't take place here. Maybe your conclusion section would be the part that comes and pulls it all together and says that. But the conclusion sections usually tend to be fairly weak. And that's not derogatory, they're not meant to be a really strong statement about the results. The introduction section is much more important than the concluding section in terms of getting the reader prepared for what is going to happen and why all this is going to come at him and how he should get his mind-frame worked so he can accept all this. The concluding section is usually just a wrap-up. They tend not to be very strong. And that's not a criticism, I think that's the way we write. All the ones I read tend to be that way, and the ones I write tend to be that way. I'm not going to make all my major points when I get to my concluding section. It's too late for that.

My papers, I think I told you this before, have almost always the same organization. There is an abstract, an introduction section with an outline at the end, a related work section where I bring in most of my references. Sometimes I cite some references in the introduction section just to lay the context and give the person a warm, fuzzy feeling that it's related to so-and-so's work, but I try to isolate more detail with that in the related work section. Section three is then the model in which I describe formally the environment and usually mathematical notation that we are going to apply here. And then what happens is one of two things. One is I do analysis. I actually do some mathematical manipulation to come up with some answers. And if I'm not able to do that because the problem is too complex, then there is just simulation results. I run a simulation...experiments using this model in certain variable settings to yield the kind of results that I'm looking for...and sometimes both. If I can do the analysis, that's good. It is very important to do the analysis because you can come up with formulas and equations that can describe the behavior in general cases, where simulations are just telling you something about one particular experiment or two experiments. So then one or both of those sections exist in terms of analysis and simulation, and then conclusions and references. My papers almost always have that exact structure.

Interviewer: Is that structure the same as all the other papers that people are writing or are some different from yours?

Billard: Yes, so it's not that different. Usually you have a related work section, so it's almost something that's an expectation. There is going to be modeling and then you are going to see results. And these results are either going to come from analysis and/or from simulations and then there are conclusions. Once in a while I have additional sections. One additional section is with substantial algorithms. Sometimes those are very simple; they are just inside the model. But sometimes there are more complex algorithms and then there is an algorithm section describing the strategies of agents in these systems.

Interviewer: In your citations, are there any rules or tabu in regard to citations?

For example, do you only cite works from ACM or IEEE? How often can you bring in completely different fields?

Billard: I bring in some other fields because my work tends to be multi-disciplined. When you do your Ph.D. they say you have to have members on your committee that are outside your department. I had a person from cognitive science and a person from economics. And both were fairly closely related to what I was doing. You're supposed to be doing a thesis that has broad applicability. Whereas the end result is, really, you are the only one who really understands it. You want to publish papers that are supposed to be multi-disciplined. The problem is no one understands multi-disciplined papers. Because wherever you submit it, the person who reads it, he's got one discipline, and you're lucky if it's even close to what you are working on. So, I tend to have broader citations, and I try to bring in ones from economics or from biology or something like that. Other people tend to have a narrower range of references and the format of them depends on the journal.

Interviewer: How about bringing in things like literature or philosophy? Is that tabu?

Billard: I have done that in papers that are NOT for computer scientists, such as the biology paper where the prisoner's dilemma is used in philosophy and economics.

Interviewer: How about quotations? Do you use many quotations?

Billard: I did that only in my dissertation. And I think that is the main place where it is appropriate because it's big...it's a book. And you want to show that you actually incorporated other research. I mean you just didn't start off from scratch.

Our first pass or no pass exam that we have to go through for a Ph.D. is a research exam. You have to take a broad view of all the research and give a presentation on it. And so, much of that then will go into your dissertation. If someone looks up your dissertation, you want to present them with a pretty substantial view of the research world there that's relevant, so I have several quotations.

Interviewer: Any politically motivated citations? I mean you want to definitely cite this person because you know that they are going to read it and...

Billard: Yes,..but also to buttress my arguments...whenever you do research you want to buttress your argument with whatever you can. And especially if you're doing something that seems to be a little bit out of the ordinary. You need to be able to put this in context...and bring in the "big guns" and say this is how they are describing this, and this is exactly what I'M trying to do. This person makes a great statement and I put it verbatim in there because this is EXACTLY what I'M trying to do here.

Interviewer: What kind of,...what kind of things have you done, what taboos have you committed that your reviewers have come back and said...

Billard: Besides the specific one about where the results have to be stated in the abstract, my main problem is the applicability. "Well, how can I use this?"

Interviewer: Yeah,...What do I do with this knowledge? Nice idea but...

Billard: They want to know what system you're going to use it on. If you did simulations, 'How does that match a REAL system?' And so that's been my major problem. Some of it has been on the exposition. I try so hard on that, but they can still come back and say... 'Is the paper clearly written and organized?' and the answer is 'Reasonably well.' And that's a pretty good response, 'reasonable.' But sometimes it's not that good.

Interviewer: Are these always native speakers that are doing the reviewing?

Billard: Well the thing is, we don't know who they are, but if you look at probability, half of them are probably not.

NOTE: This is an interesting point that Billard brings up. Most of the computer scientists in the world are non-native speakers of English and, yet, are asked to judge and offer advice on not only a paper's research results but also its English.

Interviewer: And how many reviewers, for the big publications such as the *Transactions*, are there usually?

Billard: Three. Sometimes I've even had four, I think, for conferences, but the conferences are usually two or three and the journals are usually two or three. Sometimes if it's just two in a journal, an associate editor pipes in. You can often tell when the editor pipes in. They fill out a referee report, but these reports stand out as how they are presented to you. I've had both in journals and at conferences where the editor has put in his or her input kind of wrapping up all the details of everybody else's input...And giving me, you know, their stamp of approval or what they think needs to be done from the editor's point of view. Or there are hints, like the person basically knows that you're going to get accepted because it is the editor and they are saying "When you present this, you should be careful to show this at your talk because these people won't know this."

Interviewer: So you usually submit a paper to a journal, and they probably have a list of people who are experts in that field that they send out copies to.

Billard: Yes.

Interviewer: And then you get reports directly from those people?

Billard: Well, I usually submit to the editor. Then the editor farms it out to an associate editor, and then he is responsible for finding the other people and then working it through its lifespan there. And then the referees send their comments back to the associate editor and then he sends them on to me.

Interviewer: How much revision does a typical paper go through before it actually gets published?

Billard: For a journal, a fair amount for a couple of reasons. Let's do an easy case instead. For a conference proceedings, very little editing occurs because it is accepted or rejected after the first submission. If it is accepted, the editor will say 'The paper's been accepted, but you are obligated to submit it taking into account the suggestions of these referees.' But there isn't another review process, and so you send in the camera-ready. I conscientiously try to take care of their questions. But you have to be very careful because sometimes they didn't understand it, and so they are making a statement that is not correct. And you know it's not correct. Now you have to be concerned that when they made this statement, they didn't understand something so you add a statement in there to try to clarify, but you can't take it any further. If the person read it two more times or three more times, maybe they wouldn't have made that statement. And you can't blame them for how many times they read it. So, I make a conscious effort to try to help those things. Usually they are not major. Usually they are making comments on more research that you could do. However, you probably have to cut material in order to get it to fit their space requirement. Those I take as helpful suggestions for future work. I tend to make simple modifications based upon the referee's suggestion. I might put some new figures in to make it a clearer exposition about what's going on because I know I'm going to have to make a talk at the conference. I try to think about it from their point of view...of what a novice coming in looking at this thing needs. So those are the only changes for a conference one usually.

If it gets rejected at a conference, then I have to take it much differently. Because I say, okay, I've got these criticisms. What am I going to do with it? Am I going to go to another conference? I've done that and I have to be concerned about...Did I address these criticisms well enough that the next batch will be okay with it. The thing is, you take a set of three reviewers and they are going to have a different set of requirements for it. Although I matched the last set of reviewers, I'm never going to see them again.

Interviewer: And the people at the new conference may love exactly what was rejected for another conference.

Billard: Right, right...Joe (Pasquale) always described it as a crap shoot when you're trying to get these papers in. Who's going to review them and what are the particular things they are looking for and what they like and what they don't like.

If you go to a journal, then it's a different story. For example, at *System, Man, and Cybernetics* everything you submit is always rejected. No matter what the reviewers say, it is always rejected. You always get the Dear John letter....We are sorry to inform you that your paper is not being accepted for publication. And the last one that came back, I went through and read the results. Everything was positive from all three reviewers. Everything was really good, they all understand what I was doing, they liked it, they thought this was good. Their comments were just helpful suggestions about this and that. But I didn't get in and it's because it's policy. The idea is to force another review cycle and for you to take this and then try again.

This one was actually very difficult to modify, the most recent one I sent to them because everything was positive. I can't send the same paper back to the editor and say "Everything was so positive that I just sent it back to you as it is." He wouldn't accept it; that's not going to happen. When you send it back you have to say what you did.

Interviewer: Do they review this blindly? Do the reviewers know who you are?

Billard: They know who I am. My name is on the paper...Now some of these places I've submitted to, you do have the option of submitting it blind if you really care about that. And that could happen; for me it's fine. I can understand why some people would do that, because there are blocks or cliques and you might want to protect yourself by doing it that way.

I have tried to publish in some higher level journals, some journals outside of my area even, because if I got even one of the five of those in, that would be very good. And also because I wanted to know the referees input. I wanted somebody outside my area to say what they thought of this. If it is a math paper, what do the mathematicians say. One math journal gave me the BEST set of reviews. My paper was rejected and it was a beautiful rejection, but these guys knew exactly what I was doing with this math, they knew much more about it, they understood it right away and they could describe what the significance was, etc. The thing is, they did not think it was appropriate for that journal, it wasn't substantial enough for that math journal. They thought it should be published someplace else. But they had really spent a good amount of time understanding the math and they were quite capable of doing that.

I also keep a database of my publications. I track the papers from technical reports, to conferences, to journals. Each paper has a current status, such as a submission date. The conferences and journals are coded so that I can easily refer to each paper. It's not so much the individual information but the tracking of the papers as they change form. They can take several paths. My Ph.D. work produced a thesis (Path 1) and some of the results went on to conferences (Path 2) or directly to journals (Path 3). As I said before, some technical reports dead end (Path 4) but others go to conferences (Path 5). Still others continue to journals (Path 6). There are also technical reports that skip conferences and go directly to journals (Path 7). There are some other papers that are never published as technical reports but go directly do conferences (Path 8) or journals (Path 9). The following table shows my publication track where each number is a code for a paper.

Billard's Publication Track Chart

```
          (A)             (B)           (C)             (D)
Path   Ph.D. Thesis    T. Report     Conf. Proc.     Journal Art.
=================================================================
 1 --------->8              |             |               |
             |              |             |               |
 2 ----------8------------------------->9                 |
 2 ----------8------------------------->10                |
             |              |             |               |
 3 ----------8-------------------------------------------->11
 3 ----------8-------------------------------------------->12
 3 ----------8-------------------------------------------->13
             |              |             |               |
 4 ------------------------>1             |               |
 4 ------------------------>2             |               |
 4 ------------------------>3             |               |
 4 ------------------------>4             |               |
             |              |             |               |
 5 ------------------------>17---------->18               |
 5 ------------------------>22---------->23               |
 5 ------------------------>31---------->32               |
 5 ------------------------>33---------->34               |
             |              |             |               |
 6 ------------------------>14---------->15------------->16
 6 ------------------------>19---------->20------------->21
*6 ------------------------>27---------->28,29---------->30
             |              |             |               |
 7 ------------------------>24--------------------------->25,26
 7 ------------------------>35--------------------------->36
 7 ------------------------>37--------------------------->38
             |              |             |               |
 8 ------------------------------------->8                |
             |              |             |               |
 9 ------------------------------------------------------>5
 9 ------------------------------------------------------>6
 9 ------------------------------------------------------>7
```

NOTE: Path 6, marked with an asterisk (*) is discussed by Billard in greater detail in Section 5.2.3. The papers upon which this discussion is based may be found in Appendix D.

Interviewer: Are conference papers always published in a conference proceedings?

Billard: Yes, I probably wouldn't submit to conference if they didn't publish a proceedings. In this case, I produce the camera-ready document whereas in the case of a journal, they typeset it, and all that.

Interviewer: Usually, how long are these conference proceedings?

Billard: They're usually six pages on average, sometimes seven. And sometimes if your paper is not accepted on the same level as others you get four. But these levels aren't as common unless there are so many submissions and there is only so much room. It's usually six and then you can buy one or up to three extra pages, usually at $150 a page.

Interviewer: So the first six pages are free? By getting your paper accepted you automatically get six pages printed in the conference proceedings?

Billard: Yes, but you have to pay a conference entrance fee anywhere from $200 to $325 and then you get a proceedings when you go to it. Some of these will often be published by IEEE Computer Press, and actually be advertised in various publications so that other people can buy it even though they didn't go to the conference.

Interviewer: (LOOKING AT THE CHART) Do any of these paths ever go backwards, for instance they start at a conference and then go back to a technical report?

Billard: No, with one exception, they are all hierarchical. The reason that they don't flow backwards from a conference proceedings or journal article to a technical report is that they are archived, and once you have them archived, there is no reason to go back and make them into a technical report.

Conferences are usually meant for NEW results, state-of-the-art,...and the journal is meant for the matured versions of these conference proceedings. A paper may skip the conference stage because it is either too substantial or long for a conference or there is not a conference appropriate for them, but there may be a journal that fits.

Interviewer: What type of feedback did you get from editors or reviewers about your use of the English language?

Billard: That's an interesting question. For the IEEE journals, there is a question for the reviewer: "Is the English satisfactory". I am a native English speaker and my grade here is usually "Yes", or "OK, but could use some improvement". I do not know what happens to non-native speakers. However, I have not received any returned papers with my English "marked-up" and, therefore, have not had a method for improvement.

Just recently, I received acceptances to a math journal and a biology journal. In both of these cases, the editor took great pains to correct my English. I discovered several (irritating) lapses on my part. This shows that there is a real difference in journals and/or subject areas.

Interviewer: In much of your talk, you refer to the word "significance" in terms of evaluating research work. What is it and how can it be determined.

Billard: That's a tough question because the answers lie in the beholder. Remember that computer science is very application-oriented. Significance often implies that the result can be applied in an important, already existing, area. A reviewer might describe a result as significant, if the reviewer knew a lot about the area and could see that this result extends it either in a big step in an existing direction or in a small step in a new, and interesting, direction. However, if the reviewer does not know a lot about the specific area, then it is very difficult to judge significance. That's why the choice, by the editor, of reviewers is important. That's also why it's important, for the writer, to target the best forum for publication.

5.2.3 Tracing the Subject's Writing Process

In earlier questionnaires and short interviews with the University of Aizu computer science faculty, several important aspects of professional writing practices were identified and outlined in several simple models. The writing situations within the daily practices of a computer scientist can also be far more complex. In Billard's Publication Track, only 3 of the 9 tracks flow directed from idea to research to the construction of one paper for one audience. If we ignore the doctoral thesis, a very unique writing situation, there are 5 paths where information evolved for a variety of different audiences via a variety of different genres. This aspect of professional practice is also worthy of some study.

In the following section, Billard discusses his thinking, his research, and his writing related to the generation of information and its dissemination in Path 6 to four different audiences via four papers: a technical report, 2 conference papers, and a journal article. First, he outlines his presentation, then he traces his writing from the genesis of the idea to the production of four different written products, showing further detail of the professional context within which the subject writes.

NOTE: The following account of the subject's writing was composed by the subject, himself.

Outline

I am going to describe a sequence of research, writing, and publishing that covered about two years. The end result is four papers, although I worked on many other papers during this time. I will label these four papers:

1. *Instabilities in Learning Automata Playing Games with Delayed Information* (IEEE International Conference on Systems, Man, and Cybernetics, 1994),
2. *Learning in Multi-Level Stochastic Games with Delayed Information* (Annual Conference on Uncertainty in Artificial Intelligence, 1994),
3. *Learning in Single- and Multi-Level Games with Delayed Information* (University of Aizu Technical Report 94-1-011, 1994)
4. Stability of Adaptive Search in Multi-Level Games under Delayed Information (*IEEE Transactions on Systems, Man, and Cybernetics*, Volume 26, March 1996).

There is a strong relationship between these papers, so strong that these should be considered one research effort. In the following sections, I will attempt to describe the details of the research, the writing and rewriting of the documents and the final stages of publication.

Generating the Idea

My dissertation provided the genesis of the idea behind the research. The main topics of my dissertation were dynamic group formation under delayed information in the

the context of game theory. These are the same elements that appear in these papers but in an entirely new format and with new results.

When I started my dissertation, a fellow graduate student in our laboratory told me about *learning automata*, which use very simple rules to increase or decrease the likelihood of future decisions based upon rewards or penalties received. My advisor told me about the Prisoner's Dilemma, a game in which group cooperation yields a good payoff but defection from the group yields a big bonus for the defector, at a cost to the other cooperating members. My research area was distributed computer systems where information is delayed along networks. I put together these elements for my dissertation: *Learning Automata Playing Games with Delayed Information.*

Preparing for Research

The model was more complex than this. Actually, it was too complex to give a nice clear picture of the model and the results. This led to the current stage of research. During my dissertation, I found work by Huberman and Hogg in computational ecosystems and delayed differential equations. Unfortunately, I did not really understand these well. It was only after finishing the dissertation and spending a great deal of time reading and understanding many papers, that I finally saw the light. The solution was easy: I just needed to model the delays in information by an average age of information; my previous approach was much too complicated. This opened the door to already-existing mathematical techniques. It only took ten minutes to write a simple program with learning automata using old information. I immediately saw oscillations in their behavior - exactly as I suspected.

Conducting the Research

The following months were agonizing. The ten minute exercise had shown me exactly what I was looking for; however, I needed to explain the behavior mathematically. I am not a mathematician and delayed differential equations are difficult. I found several more papers and ordered several books. Reading and understanding these references took several months. There were many false starts. I kept research notebooks of my thoughts, trials and errors, and solutions. During this time, I filled three 50-page notebooks with scribbles, math, and pictures. I performed many simulations of learning automata playing games with delayed information. This gave me general ideas about their behavior and formed the basis for double-checking my mathematical predictions. The only real writing that occurred during this stage was outlines in my notebooks. These outlines listed the main topics of the research.

Obtaining Results

Eventually, I was able to derive a formula that predicted EXACTLY how much delay could be tolerated before the players in the game exhibited persistent oscillations in their behavior. This formula was plotted on a graph and compared to the experimental results of simulated behavior. This was basis of a working paper that eventually would grow into Paper 1 and form the first half of both Papers 3 and 4. My goal was to disseminate these results to the appropriate community.

Assessing the Results

My next task was to assess the significance of the results and the appropriate forum for their dissemination. I had a fair amount of knowledge concerning learning automata which dated back to my dissertation. Learning automata had been used in a wide-variety of applications and were quite popular in the 1970's and 1980's. But NONE of the research considered the effects of delayed information. I had taken a very simple model, added delayed information, simulated many experiments, and mathematically predicted the behavior. This would be significant in any environment with delayed information; distributed computing systems is just such an environment. I was very concerned with a timely distribution to a community that would understand the results.

Selecting the Target Audience

I first chose a math journal for publication of these results. The original definition of learning automata and the conditions for their stability had appeared in this journal in the early 1980's. My results showed that learning automata were not stable under delayed information. I believed that this journal would be receptive to this new information. In addition, I had employed delayed differential equations which is an appropriate topic for a math journal.

Writing the Paper

It took approximately two weeks to write the paper. My goal was to make a very tight mathematical paper, not long on exposition. My research notebook already contained the results and various outlines. I wrote a shell of a paper with the different sections and, at the same time, I included figures and equations. Next, I wrote the abstract. Although it was only eight sentences, it took several hours. It had to be just right. I filled in the results sections quickly and then undertook the introduction. This took a full day as it is very important to motivate the problem. There was always a little voice in my head, probably my advisor, that kept saying "What is the point of this sentence?" I completed the related works and conclusion quickly. Finally, I reread the paper approximately fifteen times, making fewer and fewer modifications with each iteration.

Obtaining Feedback and Revising the Paper

After feedback from Alice Riedmiller, a research associate, I submitted the paper to the math journal. I received a rejection. The reviewers understood exactly what I was doing. They gave some very good suggestions but said that paper was not significant enough for the journal. They really hoped that it would be published somewhere. I modified the paper according to the reviewer's suggestions, but this was in the technical area, not in the presentation.

Selecting a New Target Audience

Next, I chose the IEEE International Conference on Systems, Man, and Cybernetics as a new target audience because 1) I had previously presented a paper there and 2) learning automata have been presented there before. The attendees have a broad range of interests and are quite comfortable with adaptive methods like learning automata. My paper was not about a hard-core computer science topic and might not have been received well, or even accepted, at a mainstream conference. This conference allowed me the freedom to present a new idea that was intended as a paradigm for general behaviors.

Revising the Paper

For this conference, I expanded the exposition to describe the general usefulness of my approach. I also included more experiments from my research notebook.

Submitting the Final Product

The conference required an extended abstract. This is usually a two-page summary of the full paper. On the basis of this abstract, my paper was accepted without comments. I then turned in a camera-ready version of 6 double-columned, single-spaced pages, with 5 sections, 8 figures, and 10 references. The paper was typeset according to specifications determined by the editor.

The Genesis of Another Idea for a New Audience

While reading the *Communications of the ACM*, I found an advertisement for the Annual Conference on Uncertainty in AI. I had vaguely remembered seeing a previous announcement and had thought that this was closely related to my work. In recent days, I had been wondering how my simple results, described above, could be related to dynamic group formation (my other main research topic). I had probably spent five minutes thinking about using learning automata again but this time to make decisions concerning group formation. I looked at the deadline for submission: I had two days to design the model, conduct experiments, and report on the results. This became Paper 2.

Conducting Research and Obtaining Results

I already had a simulator for learning automata playing games with delayed information. I added new code to have the automata make one more decision: should they play game A or game B. The games had different payoffs and the choice represented the decision to form a group or to work alone. Again, I ran experiments and found oscillations in the behavior of the learning automata players. These results were more interesting because decisions were made at two levels: which game to play and how to play the game.

Writing the Paper

The simulations had taken one day, I had one day left to write the paper. I would not ordinarily rush a paper like this, but I had no choice. I followed my usual style of sectioning and presenting the results first. I finished the paper as I had with Paper 1. Immediately after sending the paper by express mail, I received notification that the deadline had been extended by two weeks since the advertisement had just appeared. I took this opportunity to carefully rework the exposition and add some new figures describing the model.

Submitting the Final Product

The paper was accepted without comment except that the editor recommended that I carefully explain certain topics at my presentation. The camera-ready version was 8 double-columned, single-spaced pages with 6 sections, 9 figures, and 17 references. The paper was typeset with a format provided by the editor.

Anticipating Three New Audiences

Both Paper 1 and Paper 2 were written in a few months, but the conferences would not be for several more months. In the meantime, I decided to join both of these papers into one technical report, Paper 3, published by the University of Aizu. The target audiences would be 1) anyone I encountered at the conferences who wanted more information, 2) on-line perusers of the University's Web pages that might be interested in my results, and 3) University of Aizu committee members who might wish to review my research accomplishments in the future.

Writing the Paper

Paper 3 required very little new writing, the only real work was editing the first two papers to make sure they flowed together well. Paper 1 provided sections describing the effects of learning automata playing a game with delayed information. Paper 2 provided sections describing the effects of learning automata faced with a decision about which game to actually play. The final result was 24 single-columned, double-spaced pages with 7 sections, 19 figures, and 21 references.

It should be noted that this is the only instance I have of a technical report following conference papers. In all other cases, a working paper became a technical report. Then the report progressed to a conference and/or a journal.

Targeting Another Audience

In computer science, it is understood that conferences are meant for the timely distribution of new results to a specific audience. Journals are used to archive more mature work, to distribute it to more people, but to do so in an untimely sense. Journals often print long versions of conference papers, that is, the papers are extended with new experiments, new equations, and more exposition. I decided to submit the technical report to a journal. The report included both conference papers and additional exposition, thereby satisfying the more substantial requirements of a journal. This was Paper 4.

I selected *IEEE Transactions on Systems, Man, and Cybernetics* because 1) I had published there before, 2) part of the paper had appeared in the conference, 3) learning automata were heavily covered in this journal, and 4) the topic was of interest to the readership.

Obtaining Feedback and Revising the Paper

The returned referee reports were very favorable, except for a few minor concerns. The reviewers understood the problem and the solution. They were positive about almost all of the criteria for judgement: purpose, significance, organization, references, cogency, what was done, why it was done, scope, English, readability, presentation, interest to the general community and to an area of specialization. However, the paper was rejected.

The editor asked me to resubmit the paper after addressing the concerns of the reviewers. That would be easy. One reviewer was particularly concerned with how much of my work was new and how much was taken from previous work in learning automata. My work was a brand-new approach to a well-understood topic: I added delayed information to learning automata. I believed that the reviewer was not that familiar with learning automata and could not distinguish the line between the old and new material. I added new exposition describing the original work and my own contribution.

Another anonymous reviewer appeared to be an associate editor as he/she seemed to know what was in the other reviews. I also suspected that I knew this editor and that this editor had been helpful on previous papers that I had submitted to this journal. My main problem was that this reviewer thought the results would be more interesting if applied to N players rather than just 2 players. This is a classic review point. Many people do "small" studies but can these results be applied in real systems which tend to be very "large"? I was in trouble. The case with 2 players is very difficult to describe, let alone find a solution. If this had arisen in the context of another paper, I may have just tried to answer this question with exposition. However, in this case, I did not have much to tell the corresponding editor about my "improvements" to the paper. I needed something to change the rejection into an acceptance.

The solution lay in starting over with a new model that was much simpler but could accommodate N players. But I could not throw away the entire paper so I added a new section that began with a simple argument. The current model was too complex to expand. But if we started with a simpler one, we would arrive at the same conclusions. In addition, some simple math showed that larger groups needed to communicate more often. This was an interesting result and I hoped that it would persuade the reviewers.

Obtaining More Feedback and Revising the Paper Further

This time, the paper came back with an acceptance. Interestingly, the reviewer who did not know the details of learning automata found the new section to be a bit "premature." The other reviewer, who again seemed to have access to this review, defended it.

I was very happy. I then reread the acceptance letter more carefully and found something very disturbing. The paper had not been accepted as a full paper but rather as a correspondence. In this journal, full papers are allocated about 10 pages and correspondence only about 8 pages (but the editor prefers 4-5 pages). I had three previous papers accepted as full papers and the current paper was much better. The reason given was that one reviewer was still not convinced about the new contribution I had made to previous work on learning automata.

The paper ran to about 12 pages, double-columned, singe-spaced. I examined the paper closely for potential deletions. There were some experiments that were not crucial to the final result. I removed some exposition. The paper now ran to about 10 pages and it was just the right length for a full paper. The only way to reduce it any more was to remove the new section on N players. If I went any further, the paper would become meaningless. In some ways, it would be less than either conference paper. In desperation, I talked to my former advisor and he suggested that I present the problem to the editor. In a letter, I argued for the significance of the work in general and that one reviewer was particularly pleased with the new section. The editor granted me a reprieve: the paper was accepted as a full paper.

Submitting the Final Product

The final paper was 10 double-columned, single-spaced pages with 8 sections, 11 figures, and 21 references. The format was provided by the editor so that I could check my paper. The submission, both by e-mail and floppy disk, was then typeset by professionals. The paper would appear in one year.

5.3 Comments

So far in this dissertation, Chapters 3 and 4 have identified important aspects of context in professional computer science practice that affect both writing processes and written products produced in the computer field. The elements have been listed and described, the writing activities have been categorized according to purpose, and both the elements and the writing have been assembled in several models to show their roles in a broad picture of professional activity. This has given the reader a glimpse of the computer science discipline, its primary writing activities, the intended purposes, and the contexts within which these activities exist.

This has been followed by Chapter 5, which provides a closer look at computer science writing practices from the perspective of one representative writer. The first section profiled some general characteristics of the writer, the second section presented an extended conversation with the writer on his work and his writing, and the third section offered the writer an opportunity to explain his writing and decision-making during the production of 4 specific documents. Though most of the text requires little comment to make it understood, it is appropriate that we discuss two aspects of Billard's remarks here that are particularly applicable to this dissertation and the context of writing in computer science. The first concerns efficiency in computer science, particularly as it affects the research/writing process. The second concerns writing instruction.[14]

5.3.1 Computer Science and Efficiency

In the interview with Billard, we learned that computer science is "the systematic study of algorithmic processes that describe and transform information—the theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all computer science is 'What can be efficiently automated?' " (p. 86) and that *efficiency* appears to be a driving force motivating much of the profession's work. We also learned that the field divides itself into many areas of specialization, some ranging in size from, perhaps 70-80 to 300-400 members. The primary forums for exchange between specialists are *societies* sponsored by the IEEE or *special interest groups* sponsored by the ACM, two of the field's most important professional organizations. Most of these societies and/or special interest groups issue their own periodicals and sponsor their own conferences.

We also observed that *efficiency* seems to be a force that drives not only the field's research goals, but also the individual research and writing practices of some of its members. Billard wrote a phenomenal number of papers during the first 2 years of employment at his current cite of employment, 33 papers to be exact. This was possible only because he knew where to look to quickly find the information he needed, organized the information well in computer files and notebooks so he could

[14] It should be noted that the subject, Professor Ted Billard, was given a copy of the Comments and Conclusion sections and allowed to offer feedback wherever he felt the researcher had misinterpreted the data. The section was then revised and added to the dissertation in its present form.

retrieve it quickly, kept good records of his thoughts and results during the research process, and produced research results that would be interesting to several different audiences in several different areas. The breadth of interest and application that his research results yielded made his work publishable in several different journals and conference proceedings, both inside and outside the computer science field.

It was also observed that one series of research activities can yield chunks of information that can be separated, grouped and/or completely transformed for dissemination to different audiences depending upon the audience's specific interests and needs. As we observed in Billard's Publication Chart (p. 139) and in his discussion of specific writing activities in Section 5.2.3 (pp. 142-154), 9 different research projects yielded more than 3 times that number in technical reports, conference papers, and journal articles. Publication path 6, that Billard describes in detail, yielded two significant chunks of data that were channeled to 2 different conference audiences, then consolidated into one technical report for local administrators, interested members of the conference audiences desiring further information, and interested researchers browsing on Internet for related research. Later the information was expanded and improved for a major journal audience. Because scientific papers usually contain sections that are somewhat self-contained, these sections can be separated and recombined to make other papers without the need to write an entirely new document each time there is an opportunity to address a different audience. In the four documents that Billard describes in Section 5.2.3,[15] the following parts can be found.

#1 (Conference Paper published in Conference Proceedings)

Title: Instabilities in Learning Automata Playing Games with Delayed Information

Abstract (4 sentences)

[15] See Appendix D for copies of the actual papers.

Section 1: Introduction

Section 2: The Model with Delayed Information

Section 3: Characteristic Behaviors

Section 4: Stability Boundary

Section 5: Conclusions

References (10 references)

#2 (Conference Paper published in Conference Proceedings)

Title: Learning in Multi-Level Stochastic Games with Delayed Information

Abstract (6 sentences)

Section 1: Introduction

Section 2: Related Work

Section 3: The Model

Section 4: Experiments

Section 5: Analysis

Section 6: Conclusions

References (17 references)

#3 (Technical Report)

Title: Learning in Single- and Multi-Level Games with Delayed Information

Abstract (6 sentences)

Section 1: Introduction

Section 2: Related Work

Section 3: The Model

Section 4: Experiments with Single Games

Section 5: Experiment with Hierarchical Games

Section 6: Analysis

Subsection 6.1: Single Games

Subsection 6.2: Hierarchical Games

Section 7: Conclusions

References (21 references)

#4 (Journal Article)

Title: Stability of Adaptive Search in Multi-Level Games under Delayed Information

Abstract (7 sentences)

Section 1: Introduction

Section 2: Related Work

Section 3: The Model

Section 4: Experiments

Subsection 4.1: Single-Level Games

Subsection 4.2: Multi-Level Games

Section 5: Analysis

Subsection 5.1: Optimality vs. Stability

Subsection 5.2: Stability in Single-Level Games

Subsection 5.3: Stability in Multi-Level Games

Section 6: Large Systems

Section 7: Conclusions

Appendix

References (21 references)

The conference papers cover two entirely different aspects of delayed information, and thus, are completely different in content except for 6 citations that both papers make reference to in order to relate the work to some of the same well-known research that has preceded these papers. Two of these have been authored by the author and his former dissertation director.

The technical report that succeeded the conference papers was meant to bring the information together so interested researchers could see both the single and multi-level dimensions of Billard's research results as well as for further development to lay the groundwork for a major journal article. The following shows where the information came from to create this new document.

#3 (Technical Report)

Title: Learning in Single- and Multi-Level Games with Delayed Information

Abstract (6 sentences) – adapted from paper #1

Section 1: Introduction – adapted from paper #1

Section 2: Related Work – adapted from paper #1

Section 3: The Model – adapted from paper #2

Section 4: Experiments with Single Games – adapted from paper #1

Section 5: Experiment with Hierarchical Games – adapted from paper #2

Section 6: Analysis – original

Subsection 6.1: Single Games – adapted from paper #1

Subsection 6.2: Hierarchical Games – adapted from paper #2

Section 7: Conclusions – partially adapted from paper #2

References (21 references) – combined from papers #1 and #2

The journal article, which was based on the technical report, was expanded and modified in several ways. The following shows what percent of each section originated in paper #3 (technical report) and, thus, what percent was original to this article.

#4 (Journal Article)

Title: Stability of Adaptive Search in Multi-Level Games under Delayed Information

Abstract (7 sentences) (95% = Abst. #3)

Section 1: Introduction (30% = Sec. 1 in #3)

Section 2: Related Work (10% = Sec. 2 in #3)

Section 3: The Model (90% = Sec. 3 in #3)

Section 4: Experiments (0% = #3)

Subsection 4.1: Single-Level Games (5% = Sec. 4 in #3)

Subsection 4.2: Multi-Level Games (10% = Sec. 5 in #3)

Section 5: Analysis (0% = #3)

Subsection 5.1: Optimality vs. Stability (0% = #3)

Subsection 5.2: Stability in Single-Level Games (0% = #3)

Subsection 5.3: Stability in Multi-Level Games (0% = #3)

Section 6: Large Systems (0% = #3)

Section 7: Conclusions (10% = Sec. 7 in #3)

Appendix (95% = Sec. 6.1 in #3)

References (21 references) (14 references = #3)

Inside the cover sheet of technical reports issued by the University of Aizu is the following statement:

The technical reports are published for early dissemination of research results by the members of the University of Aizu. The completed results may be submitted later to journals and conferences for publication.

This practice seems common in computer science. Technical reports document research activity and describe results for selective distribution to obtain feedback and to claim research territory early. Journal articles, on the other hand, disseminate substantial results to a wider audience for archival and use by other researchers. Publication in a major journal may delay research announcements more than 12 months. This is a time lag that researchers on the cutting edge of new technology can't afford if they are competing with other researchers for *original* research findings. This makes technical reports with their evolution into conference papers and journal articles quite necessary. The process takes time, however; and ways to make the transition more efficient are continually sought.

The sectioning of scientific papers eases the transformation from technical report to journal article, or any other transformations for different audiences. As Billard explained in both the interview and in his own written description of his writing process, he usually begins a paper by creating the sections and then filling in the graphs, equations, or models. Next, he may write the abstract, then the results sections, and after that the introduction. After this he writes the related works section and the conclusion. Each section is an independent paper of sorts, fulfilling a different function. As observed above, parts of these sections can be efficiently copied electronically from other documents and then be altered or expanded to meet the needs of different sets of readers. Readers outside the area of specialization, for example, may require more exposition in the research methodology section than readers who conduct the same kind of experiments in their own laboratories. The related works section, for example, may require different citations for different audiences to demonstrate how research relates to different fields. The results may not change much from document to document (unless there has been additional experimentation), but the related works section and the discussion of the results may differ considerably for different target audiences.

When Billard discussed his dissertation, for example, he said, "my dissertation is not one monolithic piece of text. It actually has hundreds of pieces of text that can be regrouped as chapters or regrouped as conference papers or regrouped as journal papers...it consists of hundreds of files and diagrams and results. (p. 104)" Computer science research generates results waiting for dissemination. Billard skillfully organizes his results in notebooks and computer files so that he has a rich store of material he can draw upon for quick assembly into documents whenever an opportunity presents itself. He would never have been able to assemble a paper to meet the conference deadline he describes on pages 148-49 if he had begun the entire research and writing process from scratch.

With *efficiency* as one of the central values in computer science, many who conduct research and publish their results continually look for ways to streamline the process and, thus, increase their productivity. The quest for efficiency guides Billard's writing practices at the information retrieval and generation stages, and it guides his writing throughout the information transformation and dissemination stages as well. The ability to chunk information into sections that can be stored in the computer and then plugged into templates and modified to meet audience needs and publication restrictions is one contextual aspect of writing in computer science that is seldom possible in the humanities.

5.3.2 Writing Education

Another topic in Billard's responses worth discussing in this dissertation is his writing education. Billard's writing instruction as an undergraduate consisted primarily of reading British and American literature and then writing critical essays in response. Even though his English courses were specifically for engineering majors, he studied the literature and writing genres of the humanities. Though Billard thoroughly enjoyed these courses and even took an extra semester, it is difficult to see how this instruction prepared Billard for writing as a computer engineer. Granted, it can be argued that his thinking and language skills were sharpened,[16] but the context of writing in his literature classes was so very different from that in his science courses, his graduate work, and in his work in industry and academia, that one wonders if this method of writing instruction really enabled Billard to become the best writer he could be. In fact, Billard claims that some aspects of his instruction even hindered his success as a writer in the sciences. In the interview on page 127, Billard implies that his literature background trained him to entice his readers into his papers like the "Dance of the Seven Veils." His dissertation director had to break him of this habit. Language in computer science papers is just as important as that in literature papers, but the cultural expectations of the two discourse communities that produce them are vastly different. Computer science papers must forecast what is ahead in the paper with no surprises. Even at the paragraph level, Billard's dissertation director continually asked Billard to put the strongest sentence first, not last. First there is an assertion, then there is support–not the reverse. As Billard states, "there are a lot of [complex] things going on" in technical papers, and "you need to really give signals to your reader...here's where we are at and what we are going to do." Particularly in computer science where so many nonnative readers of English are present, the message must be obvious; otherwise, chances for misunderstanding are dangerously too high.

The bulk of writing input that appears to have served Billard the best has been from his dissertation director, a man who mentored Billard throughout his research

[16]This is a frequent claim in literature departments that claim their instruction has value for students of all majors.

and writing apprenticeship. This, you will recall, is how most of the other computer scientists surveyed in Chapter 3 got the bulk of their training as well. As Billard states on page 98, his professor "really cared about quality writing." He spent hours with Billard "grilling" him on his language and the presentation of his results. In a casual conversation not recorded in this chapter, Billard once shared how he had happened to be flying on the same plane with his mentor one time and that they had discussed writing the whole hour or so of flight time. Billard's professor not only read Billard's papers and provided helpful feedback, but he also helped Billard find channels for publishing his research and co-authored several papers to guide him through the research and writing process. Now, three years later, Billard and his research mentor, Dr. J. Pasquale, are peers. Both are university professors heading research laboratories and actively publishing their research results. Now they give each other constructive feedback on research papers.

The other rich source of writing instruction for Billard has been editors and reviewers that respond to papers he has submitted for publication. Those within the computer field have generally offered advice on the content and general matters of presentation. Those in other fields, i.e., biology where Billard's work also has application, have given feedback on Billard's language. Billard saves the comments he receives on his papers and uses them to guide the writing of future papers. Occasionally, he sends off a paper to a very competitive journal knowing that his chance for publication there may be somewhat slim. If rejected, he may lose up to a year of time as the paper goes through the review process. The comments he receives from reviewers of top journals, however, are particularly valuable. This quality of input can significantly improve his research and his writing. As he explains on page 137, some very good mathematicians gave Billard one of the best reviews on a paper that contained a lot of math. It was a top math journal, and Billard, not being a mathematician, wanted to see how mathematicians would respond to his work. There was a chance that they would publish his paper since they had recently published an article on the same topic. Billard's paper took the topic further. But if they didn't, the input would be well worth the time and effort invested. Later, the paper was, in fact, rejected. The input Billard received from the mathematicians, however, made the paper much stronger for eventual publication in a good computer science journal.

5.4 Conclusion

There are clearly other important issues that Billard raises in this chapter that are worthy of study. For example, there is the matter of how he personally adjusts his language and presentation to meet the expectations of each journal and conference he targets. There is also the matter of how he interprets and responds to reviewer comments in his papers. There are other issues as well, but these must be studied later. The initial research phase necessary to investigate writing practices within an academic discipline that will lay the groundwork for future research and instructional applications must limit itself to identifying the major elements of professional practice within which writing finds its natural home. In Chapters 3 and 4 computer science faculty identified a number of elements in the writing context that enabled the construction of some general models of writing practice. In Chapter 5, we see that this process is primarily learned, at least for Billard, through mentoring from research advisors and from journal editors and/or reviewers. (Of course, the almost daily reading of computer science genres also contributes to a writer's training, as suggested by responses from earlier questionnaires.) We have also observed that *efficiency* motivates much of the research and writing activity within the computer field.

This influences the way computer scientists store information and assemble it into documents to meet the expectations of specific audiences.

These are the major influences that affect Billard's writing, and most likely, are fairly typical of those which influence the writing of other computer scientists as well. Deeper investigation of Billard's writing may yield interesting results, but that must wait until more research has been done to find which practices are common enough to be worthy of the investment. The next logical step, rather than going deeper, must be to investigate how a nonnative writer of English writes professionally in the computer science field. Since the majority of computer scientists do not speak English as their mother tongue, it is necessary that we compare Billard's writing practices with those of a writer who brings cultural and linguistic diversity to the professional writing context. This is done in Chapter 6.

Chapter 6

Case Study: Nonnative English Writer

In order to obtain a fuller picture of writing in the field of computer science, it is necessary to investigate the writing practices of a professional who did not learn English natively, particularly since more than 50% of the computer science population is composed of NNSs. Lacking the *native-speaker advantage* means that the process of obtaining information from professional publications or conferences, most of which employ English, and the process of disseminating research results take more time, more energy, and perhaps more training to do as well as native English writers with equal research talent. How do NNWs deal with the language handicap? What are their strategies for coping? The following chapter shows how one NNS responds to the added challenges.

6.1 Methodology

6.1.1 Selection of the Subject

The subject for this study was selected according to criteria very similar to that used to select the native speaker in Chapter 5, the only differences being number 1 and number 6.

1. The subject was NOT a native English writer.
2. The subject was male.
3. The subject was active in research and writing.
4. The subject was highly conscious of his writing.
5. The subject was enthusiastic about participating in this study.
6. The subject was similar in age, rank, and professional experience to the native English writer studied in the previous chapter.

After reviewing responses to the preliminary inquiry of non-native English writers in the computer faculty at the University of Aizu, it was decided that Dr. Kyung-Goo Doh, Assistant Professor in the Department of Computer Software, would be the most suitable for the case study of a NNW.

6.1.2 Method of Investigation

The method of investigation for Doh, consisted of the following steps, explained in brief, below:

1. Isolate the subject's responses to previous questionnaires from each electronic file organized by questionnaire number.
2. Print out the responses for assembly in a ringed notebook on the subject.
3. Use the data to create a *Subject Profile.*
4. Send a copy of Chapters 1-5 to the subject to familiarize him with the context of his participation in this dissertation.
5. Send a copy of the *Subject Profile* to the subject for him to revise and expand if desired.
6. Send a copy of interview questions to the subject to enable him to think about his answers and prepare any illustrative material necessary to supplement his responses.
7. Conduct and record an interview.
8. Transcribe the interview tape.
9. Identify important segments of the conversation relative to this research and organize it for ease of presentation.
10. Analyze the data from the subject and comment on the findings.

6.2 Results

6.2.1 Subject Profile

Name: Kyung-Goo Doh

Country of Birth: Korea

Native Language: Korean

Education

1992 Ph.D. in Computer Science (Kansas State University, Manhattan, Kansas)

1987 M.S. in Computer Science (Iowa State University, Ames, Iowa)

1980 B.E. in Industrial Engineering (Hanyang University, Seoul, Korea)

University Rank: Assistant Professor

Department: Department of Computer Software

Research Lab: Language Processing Systems Laboratory

Area of Specialization: programming languages

Professional Work Experience: Doh has had eight years of experience as a university researcher and educator.

Language Education: Doh had six years of English instruction in a Korean junior and senior high school. The focus of education was primarily on grammar, translation, and reading skills. At the undergraduate level at Hanyang University, Doh had a one-semester course in English literature and writing instruction in a course offered by the English literature department. Upon graduation, Doh went to the United States where he learned technical writing from a course exclusively designed for nonnative speakers. In that course, he was taught how to write technical articles in English. Since then, he has slowly improved his writing by reading many technical articles and by receiving feedback from his advisors and colleagues.

Writing Production: Doh has presented and published four conference papers; published four technical reports; published one journal article with a colleague, and has two journal articles currently in preparation.

Target Channels of Dissemination: Doh stated that the following professional journals are the most suitable for disseminating the kind of information his research generates: *ACM Transactions on Programming Languages and Systems*, *Journal of Functional Programming*, *Theoretical Computer Science*, *Science of Computer Programming*, and *Computer Languages*. In addition, he stated that the most suitable conferences for disseminating his research results are the ACM Annual Symposium on Principles of Programming Languages (POPL); ACM Symposium on Partial Evaluation and Semantics-based Program Manipulation (PEPM); ACM Conference on Functional Programming and Computer Architecture (FPCA); ACM Conference on Lisp and Functional Programming (LISP); European Symposium on Programming (ESOP); and ACM Conference on Programming Language Design and Implementation (PLDI).)

Professional Societies and Special Interest Groups that Relate to Area of Specialization: IEEE Society on Computers, ACM SIGACT, ACM SIGADA, ACM SIGAPL, ACM SIGPLAN, ACM FORTRAN FORUM, ACM LISP POINTERS, and ACM SIGSAM.

Experience as a Referee: Doh has refereed papers for the *Journal of Functional Programming* and for the conference *ACM Symposium on Partial Evaluation and Semantics-based Program Manipulation.*.

Writing Difficulties: Doh believes that scientific papers are the most difficult to write; however, all English can be difficult.

Writing Expectations for His Students: Doh would like his students to be able to write technical reports describing their research results or findings. These should be accompanied by oral presentations.

Instructional Expectations for the English Language-Teaching Faculty: Doh stated that the CLR can aid students, by improving their skill in reading, writing, and presenting. For the nonnative faculty, Doh would like to see the CLR assist professors improve their writing by reviewing and commenting on the articles they write before they send them off to be considered for publication.

6.2.2 Subject Interview

The following conversation took place in the researcher's office in June of 1995. The conversation has been transcribed and subtitled to facilitate comprehension.

Area of Specialization

Interviewer: To someone like me who is not a computer scientist, how would you explain your area of specialization?

> **Doh**: In one phrase, *programming languages*...and then in more detail, programming language design and implementation.

Interviewer: I noticed on your Web page the term *action semantics* appears quite often. Could you explain what this means?

> **Doh**: It's a formalism to describe meaning of programming languages, including programs too. For example, in natural languages, take English for example, we have syntax which can be defined in some formal way. But the meaning of the English structure is very hard to define. I mean it's maybe impossible to define because there are a lot of ambiguities. But fortunately programming languages are much simpler than natural languages, so we can...we believe that we can formally

define both syntax and semantics. So, in computer science, syntax has been studied and well established. And so we have formal tool to define syntax. But semantics has not been that easy. But, more recently a lot of people worked on it, so the progress has been made. There have been many different ways of defining the semantics of programing languages. Action semantics is one of them. It's kind of most advanced and mathematically well founded. And also it's practical, so that we can formally describe real life languages people are really using now. When we write a language manual, we use it to describe the meaning of program phrases in English. And there can be a lot of ambiguities, and also uncertainties. So it causes a lot of confusion. But now we have some formal notation or mathematical tools to formally define the thing. And then it causes no confusion. So that's one goal of having formal semantics. We can describe meaning, formally.

Another advantage is that we can automatically derive system implementation from formal definition. Moreover, the automatically generated compilers can be proven correct, and can also compile programs consistently.

Interviewer: When you conduct research, what kind of results are you looking for? What is the product of your research?

Doh: Well, one thing is...try to develop a model of the thing you want to do. Say for example, I like to develop a system which takes formal semantics of a programming language and then it gives me an implementation of the language, I mean a compiler of the language, for example. First, we should consider what kind of semantics method we want to use to achieve that goal, and then, how we can make it efficient. There are a lot of things going on to achieve that goal.

Interviewer: Do you usually work alone on your projects, or do you work with someone else?

Doh: For my Ph.D. research, I have consulted with my advisor. But after that, it happens that I just work alone. I have no one around here working in the same area, so the only communication I can have is with some of the researchers in the world who do similar research, who do some work in this area. And have a close relationship with them, and then we mostly have e-mail communication and also we meet each other at the conferences.

Interviewer: Who is the primary audience for the information that you are creating?

Doh: The people who belong to this group, for example, SIGPLAN...most people who are interested in programming languages, for short. Many people are interested in semantics, or some are interested in the implementation of the programming languages, or some people are interested in studying the properties of programming languages. But they are all somewhat interrelated.

If eventually, this area is successful, the audience could be students and software engineers who are working in industry. They will probably get affected by this. But at the moment, only the specialists.

Interviewer: How many people would you estimate are in the same research area?

Doh: People who are working on the exact same thing with the same goal is about 10 people at around 5 different places. The general goal, maybe 100. But using action semantics, only about 10. But we are very cooperative. We don't compete with each other. That's very good.

Writing Practices

Interviewer: What are the most common documents that you produce?

Doh: Normal e-mail is the most. Usually, that is with these 10 specialists or other specialists. Some of it is casual, and some of it is technical communication.

Interviewer: I notice that you also write technical reports, and conference papers for conference proceedings, and journal articles. Could you tell me a little bit about these?

Doh: The main purpose of technical report is kind of intermediate medium to present what you have done. So you have some idea, and you have some results, and then you just write down what you have. The medium of technical reports can let everybody know that you have done something. That's one purpose of technical report.

Then another purpose can be...for example, I have conference paper published in the proceedings and then...well, I can have some extensions or minor revision or major revision and I might want to publish it as a journal paper. Also, you can publish it as a technical report before it appears in a journal so you can show it to other interested people. That's another purpose.

Interviewer: And who reads your technical reports?

Doh: Well I belong to several electronic mailing lists, and I know a few of the subscribers to the mailing lists are interested in my work, so I announce it to the mailing list and then people who are interested will send me an e-mail and then ask for the report and then I send it to them.

Also there are some experts I know, and from whom I need feedback, so I send them my technical report then hoping that I can get some feedback or criticism or whatever. Or sometimes I will ask them for some feedback directly if they have some time....but AFTER World Wide Web, people can access my Web page and the Action Semantics Web Page...now we put all our stuff on that page. It is a collection of all the papers in that area and there is one maintainer, and also there are postscript files of all the papers. So now I send all my papers to the maintainer and he keeps them in his disk, so everybody can fetch it from there. It's very convenient, very fast when we want to do research. If somebody has done something that I am interested in, I can fetch it and then directly print it on my printer. It's very efficient. So it is quite amazing to see what has been happening these last couple of years.

Interviewer: What about conference papers?

Doh: Well, now it's kind of different, because up to a few years ago, there was no concept of this kind of Web. We had some ftp directories where we can fetch papers. So conference papers, were originally—-and now still because Web is still not widely spread—-were to share information very quickly. We meet once a year, and everybody has their research and they present their research, and then we have communication that way. But now, the last couple of years, it's not that way because communication has already been done. So by the time I give a presentation, it's already obsolete. I mean we have had the communication already by e-mail and the Web page. Everybody already has my paper, so it's just kind of a formality. I recently had one conference paper accepted. It was very competitive and so by going to the conference, I can say I have done some good work and that's it. And so conference paper is becoming kind of formality. And then at the conference, people meet and then we talk about future work. But the original purpose of conferences was fast communication.

Of course conference is a very good way of knowing people. These days, we get to know people by e-mail, I mean even become good friends. But at conference, we can meet each other face-to-face. So conference is still very important.

Interviewer: Have you ever co-authored a paper and done all the writing by e-mail?

Doh: Oh yeah! Once with my former advisor, he had some idea. And then he sent it to me. Well actually, we didn't have much time. He sent me his memo–about a 10-page memo. And then I worked from there and finished it. And then we met at the conference, it was really more of a workshop, and presented it. It shouldn't work like that, but we didn't have much time.

Interviewer: How about journal articles?

Doh: Conference papers and technical reports are sort of intermediate. Some conference papers are good enough to be final, but normally journal papers are finalized versions of everything. You have done some work, or you have discovered something. Well it should be published permanently as a record of work that has been done. We can record it as a journal paper. It's kind of permanent record.

Interviewer: I wonder if you could lead us through a typical writing sequence for you, tracing a project from the idea generation stage to the final publication.

Doh: Well, I have written just a few papers, and it might be different in the future, but from my limited experience...First, I have to think of the general direction, I mean what kind of achievement I want to do. It normally happens when I meet someone and talk with someone or read someone's paper. I think, 'Well, if this can be done, it will be great.' And then I try to sort out the problem, and then maybe a note or some scratch papers, writing some formulations or drawing pictures and things like that...try to solve the problem. And then, I write sort of a skeleton paper. I try to write down the idea, of course in English...I don't write in Korean because I started this thing in English so it's very hard to switch back and forth. So even if I think in Korean, when I write, I write in English. Then when I first finish skeleton, I try to write some sentences in the skeleton from my memos or things. I type it in the computer so it can be permanent. And then, finalize the problem. If we have to show something that the formulation is really correct or not, you have to write down all the formulations and try to think if all these formulations are really okay and then I type them in. So I try to fill in all these gaps I have in the skeleton and gradually complete the thing. And then once the idea is written down very roughly, I mean the introduction is empty and the related works are filled in very roughly and the formulations and some memos in the different sections in the LaTeX formula, then I print it out and look at it. And then add more information, and print it out again and then fix it and then add something, if there is something new or needs to be corrected. Once the formulation has been done, I write the program to see if it is really implementable or not. Writing down programs helps me better understand what's going on. So you can convince yourself that what you are doing is really doable and okay. And then also by writing programs, you find something wrong in your formulations, normally. So you can fix those things by writing programs. And then when you think the research is done, then I add the abstract, introduction, and conclusion, and the related work—what have I accomplished in relation to other work. Normally, by the time I'm finished writing this, I'm approaching the deadline so I send it out just before the deadline.

Interviewer: Do you ever get any input from others on your writing?

Doh: Usually not, because I have no time left before the deadline. Well, sometimes if I have some time left, I send it out to some friends, some colleagues in the world, maybe five or six people. If there's time, I can get some feedback. And most of them give me a lot of good feedbacks. So I put those things in the paper.

Interviewer: So writing the paper takes so much time, that when you are finished, there is seldom enough time to get any feedback from others.

Doh: Right. That's right.

Interviewer: When you send a paper off to a conference or journal, what kind of feedback do you usually get from your reviewers or from the editor?

Doh: When I wrote papers with my co-author, I didn't have any comments on the English. He was a native speaker and he could make the English good. But when I have done papers alone,...well normally people don't give comments on English. If they are confident in English, they say "English can be better." That's it. But I don't know how to fix things. Usually they just give technical comments. It seems the technical comments are more important when you are given a very short time of reviewing. So normally, for conferences, referees have at most three weeks and maybe 15 or 20 or 30 or even 40 papers to review. So they really don't have enough time, so they try to distribute the papers as much as they can to the people you know who can be good referee. But those referees still don't have much time. The comments I got is mostly technical.

Interviewer: Now I notice that on the referee reports, that I have collected from ACM, IEEE, and various conferences, that on almost every report there is a place where referees are asked to evaluate the English. But when you get these reports back from the referees that respond to your papers, they may say something about the English, but they don't tell you specifically how to make corrections. Is that right?

Doh: Yes, that's right. In my case, most of these feedbacks is on conference papers for proceedings. My journal article has been written with a native speaker. He can fix the English so I didn't have to worry much. I believe his English will be better than mine, so if he changes my way of writing,..well I have to believe him. His way should be better, so I didn't get any comments from reviewers or editors of articles on English.

Interviewer: When you work on a project alone, you have this idea, then you start with the mathematics? Is that where you begin?

Doh: Yeah. There are some mathematics we use to describe things and also the reason about things.

Interviewer: So then you start with these formulas,...now are these done on paper or on the computer?

Doh: Normally on paper.

Interviewer: Then once you come up with something that you are satisfied with, then do you start writing or do you go to the computer and start putting down these formulas?

Doh: Yeah. I put these in the computer. Then I think about how to organize the paper, and put in the skeleton, and make sections. But sometimes these change a little as I write. I don't think there is always the same pattern. It starts with the ones I am more confident with. I do those first. If I have a good idea on related works, then I start to do related works. But I think for the native speaker, have some memos and then they just sit down and write the whole thing from the beginning from the introduction. It seems that way. But I cannot do it because my thought processes are not linear...my case is I have to write a lot of things, whatever I have in mind, and then I have to change a lot. I have an image in my mind what I want it to look like, but to get there takes a lot of trouble. I try writing something, what's in my mind, but...I'm not very logical in presenting thing in English, probably. I have not been trained that much so it's kind of difficult to write things in one path. It takes a few,...several paths to get to the stage I think it's okay.

Interviewer: Do you do a whole section and get that perfect first and then move to another section?

Doh: I usually have many sections going at the same time. This order depends on my confidence. Normally, I never satisfied what I have written, because I know something's not just quite clear. But, I just cannot write the things in my mind. Even on e-mail messages, I cannot write in just one path. Sometimes, I can do it, but usually I have to go back and erase some things and then reread again and then, 'Oh, this is not right.' And then change it. It is like a mosaic. I reread and reread and change it and change it. I think somewhere in my brain I think I have some kind of ability to judge if it's good writing or bad writing. It's not perfect, but somehow,..I think it's from my experience reading a lot of things. We read so many articles and things in computer science. I can judge kind of intuitively, "Ah, something's wrong," but I don't always know exactly what it is.

So sometimes after several days, when I'm in the right mood, I can write better and sometimes I see, 'Oh this can be better.' and then I change it and it looks better. So it just takes a lot of paths through the paper to make it really good.

Interviewer: How are you making these revisions? On the computer?

Doh: I do both. Sometimes on the computer, but I think more effective if I print out and read it on paper. I think I can concentrate better when I look at things on paper.

Interviewer: Now some of your papers have been written jointly with a native speaker. Could you explain how the writing process works with two people?

Doh: It starts with a discussion. And then we agree on something and then,...for example, we write similar formulas and then discuss how to prove the thing and how to make the thing correct and things like that. And then, if you agree on things, you can start writing. Most of the work I have been doing, we agree on things, and then I write the whole thing, and then give it to him and he changes things and then he add things and correct things, and come back to me and then I do this and back and forth, back and forth until we finally make it. I usually do the first draft. Some people may divide the sections and each do different sections. I think some people may do that thing, but I have not.

Interviewer: What is the skeleton that you usually follow? Is it always the same or does it differ from paper to paper?

Doh: There is always an abstract and an introduction and a conclusion and some kind of related works. Sometimes the related works goes after the introduction and sometimes it goes later. And then the main thing I am doing. In my papers, I don't need a methodology section like in biology or chemistry. Normally we don't do it. Just the results....the conclusion is normally a brief summary of what I have done, what I have accomplished and then future work. The discussion is usually in the results sections and sometimes in the conclusion too. Maybe each results section has a different results or some example of results. Sometimes there is no conclusion.

Interviewer: Sometimes you have information written in one paper than you might want to transform for a different audience in another paper. Do you have any examples of that process that you can explain?

Doh: Well, I have one conference paper that I transformed into a technical report. It was a vast extension of the conference paper. In a conference paper you only given around 10 pages for the proceedings, but the research may need to be recorded on 20 or 30 pages for somebody that want to know your work. In my technical report, there is a lot of development in each section and add some new sections and formulas.

Sometimes its hard to say a paper develop into another paper. The idea in one paper start the idea in another paper, and then they become completely different papers. Sometimes the emphasis is completely different.

For technical report, some universities subscribe to a university's technical report and so researchers there look at the title and see if there is anything that interesting to them. That's one purpose of technical report. Also sometimes after you submit a paper to conference proceedings or journal article, you want to make a technical report to give to people while waiting for publication.

Interviewer: You've refereed papers for conferences and journals before. When you respond to a paper, do your respond only to the content? Do you ever respond to the language?

Doh: I respond to the content. Sometimes I see something in English is really lousy, but I can't tell them how to make it better. I can correct grammar; I have done that but I can't give them any more details other than grammar. I can tell them to add an 's' or something, obvious things, but not more than that. The writers are Danish or European and their English's pretty good. But sometimes something's wrong too. I know when I speak, my sentence is sometimes wrong but I can't correct that because it's too late. But correcting grammar in papers is easier.

Some nonnative speakers are quite good with English. I had one occasion given a paper to a friend who was a nonnative speaker, and he gave me lots of feedback on the writing style. He gave me lot of advice on abstract and other things. But I cannot get the same feedback all the time because everybody is busy. But on one occasion this guy gave me lot of feedback.

Interviewer: As a nonnative user of English, what is most difficult for you as you use English in your work?

Doh: I think finding the right word is very difficult. Sometimes I know what it is in Korean, and I know the word I pick in English is not quite right, but I cannot find an alternative. So then I have to spend a few days and think about those, and I also don't have the right dictionary,...I don't know. But sometimes I find a word very close, and sometimes I never find it, and sometimes I cross out the whole sentence and change things so I can avoid that word. Of course all the aspects are difficult and takes a long time, but finding words are maybe most difficult. More or less, I can judge the quality of the English, until I accomplish that level, I have to rewrite and rethink. It's not like just sit down and write the whole thing. I write little parts and then reread the next day and rewrite and think again.

I think native speakers do this way. Maybe they write the whole thing and then come back and rewrite some part and do that several times to make final version, but I have to do it maybe a hundred times.

Interviewer: I noticed in your papers that sometimes you use "we" even when there is only one author. For example, you say, "we introduce" or "we studied," but you are the only author.

Doh: Yes, it doesn't matter if it is one author or two. I always use "we." I learned it from other papers; everybody's doing it. In one paper I noticed "I," but it was an opinion paper. He presented his opinion about some matter, so I think it is right. But in most other papers, hundreds of papers, even if I don't read it, I just look at it, they use "we." If I say "I" it is just too strong. Sometimes people want to avoid attack from others so they use "we." Also "we" means generally this thing is true. "We have done this" means everybody can do it. It is kind of general truth or general thing that can be done by everybody.

Writing Education

Interviewer: How did you learn to write? Can you tell me a little bit about that?

Doh: Oh, it is horrible. I first learned English when I went to middle school, I mean that's when Korean students start English. We started from ABC. But I think the education system was not quite right. Mostly the focus was on grammar.

And then we all studied English very hard to get a high score in English exam. I think I studied hard for English because I liked English so I studied some. I mean my effort was okay. The result was okay. I think I had a good score in English. But even after completing high school, I had never spoken a single sentence. This is true. When I met an American or English speaking people, I can't say anything other than 'hello.' That's the only thing I could speak. I never practiced. But I was able to read it and comprehend any English sentence, even very hard sentences like President Kennedy's address. I can comprehend all the things. But English writing or English speaking was the thing I had never done.

In college, we had I think one course for English writing and reading. But mostly, we spent the most time on reading very difficult English literature. And then we wrote one report about some topic...I don't remember because I didn't spend much time on it. I think it was a couple pages of written report about something. The professor was Korean and from the English department. So I didn't learn much, I think. But then, I was not very serious about English writing at all.

Interviewer: Now your undergrad degree you earned in Korea was in industrial engineering. Then you went to Iowa State University for a master's in computer science. What happened there with your English?

Doh: I studied before I went to the States. I studied only for TOEFL. And I got an okay score on TOEFL, but my listening was very bad. I got a very good score on the other sections, but the listening was terrible. I got a total of 590 or something, however. But even on the plane to the States, I cannot even ask the stewardess to bring something. I couldn't do it.

Then, after I arrived, there was an English exam for the new arrivals. So I took it and I passed all of it except writing. So, I had to take a course for foreign students in writing. I think I learned everything from there. That's the only education I had, real English writing education. I think the teacher was very good. I became very close to him. He was excellent. During the course, we had about 7 or 8 reports, from short reports to kind of a final paper. It was a very technical paper. We learned the rules like what kind of thing we have to put in the introduction and in conclusion and things like that. And then we had to present the report. We had to make some slides and then so I think I learned a lot. And I wonder if I ever succeed without that specific one-semester English writing. And that helped me a lot.

Interviewer: What kind of topics did everyone write on?

Doh: Mostly everyone wrote about things in their major.

The instructor, actually, was a graduate student in TESOL or something like that. So it was really helpful. Then I didn't have a chance to write anything until my master's thesis. We we had to hand in a lot of projects, and on the exams we had to write down the answers. But as time went on, I realized that without that education, I probably had a hard time getting over these things.

Then gradually I learned how to speak and my listening improved a lot. So that's the only education I got actually. Later I became a TA, and the university asked us to pass an exam for the TAs. But that one, it was mainly speaking and answering questions. I failed it, and I had to take a course for TAs. The course was mainly for fixing your pronunciation. There was a therapist, and she was experimenting with me. (We laugh.) So it was really fun, but I don't think it helped a lot because she was mainly interested in correcting my pronunciation, and also she was gathering information on how Korean has difficulty with some kind of pronunciation, and how Japanese have difficulty with some kind of pronunciation. So I learned that what kind of vowel I am pronouncing is not really right. But I was really busy for my thesis than learning English pronunciation.

Interviewer: What was your thesis topic?

Doh: The title was *Generating Standard Representations from Pascal Programs*, but it was kind of....well, when software engineers want to gather information about programs of real software, normally softwares are confidential, so companies don't want to give their software for the software engineer to study. So, I developed some kind of translator that extracts all this needed information from programs but still hiding the real semantics of the program. So companies can give the software engineers the information they need to study without revealing their secrets.

Interviewer: When you wrote your thesis, what kind of help did you get?

Doh: It was totally up to me. My advisor was an American and he corrected a lot. So he was the only one who helped me. All the writing was done by me, but he was correcting things and moving around things...he worked mostly on the grammar and organization, most importantly the organization. So when I look at his correction, I say, "Ah! This should be better." And then I corrected it and things like that.

Interviewer: And then you went to Kansas State right after that for your Ph.D. What kind of writing did you have to do during your Ph.D. program?

Doh: Well, I have to take another exam for TAs because Kansas State University had different rules. Then in my graduate classes, normally I had to write a report, a kind of paper or technical report. But other than that, I didn't have much writing until my fourth year of my Ph.D. At the time, I finally got something. But for the first three years I was just struggling to learn things and also try to find the topic. I was hunting for some topics, but I failed to progress it for a couple of things. So, I think fourth year I finally got one that seems to be good.

Interviewer: When you began writing your dissertation, what kind of help did you get?

Doh: I normally communicated with my advisor. Actually we were writing a paper in 1991, and then he corrected me a lot. I think all the major sections are done by me, but a lot of discussions with him. And he corrected me and he wrote the introduction, and he wrote the conclusion and also he corrected me a lot, also the way of presenting the things. I mean, formulations and everything. And then, actually then my thesis started from there. Mostly done by me. He was very busy, so he didn't really corrected much for my thesis. So, I don't know if my thesis was really structured right or not. He didn't give much comment on the thesis. But I think the starting point was this. This was a good starting point because I learned a lot of the way of writing and also I found many of my writing style was not really good. Sort of like, I learned a lot of his writing style. And I found that his style is kind of different from others too, but I just had no choice of learning his style because, I know his style is okay too.

Interviewer: What kind of problems did he comment on most frequently?

Doh: Well it was...sometimes he corrected, well for one section, he suggested the completely a different way of presenting the same thing. So we had to completely erase everything there and rewrite the whole thing for a different presentation. I think three or four times we went back and forth to make a better presentation for the same thing.

Writing Required of his Students

Interviewer: What kind of English writing do you want your students to do in the undergraduate program or in the future graduate program?

Doh: I will ask them to write short paragraphs explaining things or describing things, like in essay exams. And then maybe technical reports or things like that.

English Language Support Desired by the CLR

interviewer: What can we, as an English support program for the computer science departments, do to help the students or the faculty here?

Doh: I think for the students lots of practice in writing and presenting things. I think that is the most important—writing and presenting. For the nonnative speakers, especially the Japanese and the Koreans, we are weak in English because the language is quite different. The thought process is really hard to adapt to the language like English. And also the cultural background is really different. They are really shy and they think that if they are not perfect, then they never try.

And for faculty, we need some support, especially for me. I think I have hit the wall. Without any support on writing I cannot improve. I just keep writing the same way. But if I can get some feedback from English expert, I learn something. Maybe with more feedback I can write more papers and much better papers than now. I think most faculty need that support.

6.3 Comments

In the interview with Doh, several things surfaced that are worthy of comment. First, it is clear that Doh has received far less instruction in writing than Billard has and that even now, the amount of feedback he receives on papers is less than satisfactory for him. He has had only two semesters of writing instruction at the university level compared to Billard's 5 semesters, and his Ph.D. advisor provided far less instruction in writing than did Billard's who seemed to have a real passion for good writing. It is true that Doh received extensive feedback on his master's thesis, and on a paper for publication that would serve as the basis of his dissertation, but the actual dissertation process provided little of the writing mentorship that he needed to make the kind of improvements in his writing that he desired.

Currently, the primary source of writing instruction for Billard and Doh comes from colleagues, reviewers, and editors. Billard continues to get feedback on both his research and his writing from his former dissertation advisor, now turned peer; and Doh continues to get feedback from his dissertation advisor, also turned peer. Both not only get feedback from their former advisors, but publish with them too. Apparently, working relationships that begin in graduate school can continue on into professional life for some time. Once apprenticed into the culture, research, and writing of a particular area of specialization, a computer scientist joins a community of specialists that work together and support each other's work throughout the length of their careers.

Another point worthy of comment concerns Doh's writing process. As a nonnative writer of English, Doh must exert far more time on his writing than Billard. Billard may finish a paper, review it several times, and then send it off to his former advisor in California or a colleague in his lab to get feedback and then revise it further. Doh also wants feedback, but by the time he finishes a paper, little time remains before the submission deadline to afford the luxury of input from others. English is not easy for Doh. Finding the right words to express his thoughts takes lots of time and effort. He continually revises and revises, intuitively knowing that something is wrong, but not always knowing how to fix it. Much of his knowledge of the technical report, conference proceedings, or journal article genres comes from reading hundreds and hundreds of these kinds of documents—all in English. He has developed a sense for what is *good* English writing and what is not, but knowing what makes something good and knowing how to transform his research results into good English is still quite troublesome. If fact, when he reviews a paper for a conference or journal, he can often tell that something is not right. He may comment on the content of another's paper, and he may venture to fix obvious grammar errors, but seldom can he go further and recommend to the author how to improve trouble beyond the sentence level.

Occasionally, he misinterprets the English rules he picks up from his scholarly reading. His use of *we* to refer to himself and to lessen his chances of attack on his ideas is one good example. He sees *we* in papers, notes that they are authored by one person, and then assumes that this is scientific convention. Because he often must skim papers, in order to cover the vast quantity of information he needs for his research, he observes the pronouns, but doesn't have the luxury to stop and analyze their use. He never realizes that *we* is frequently used by a single author to refer to himself and his readers as he leads them through a paper, as in "we have just observed in the previous paragraph" or "next, we will consider the following formula." With little time to consider such matters, and with little time for input from knowledgeable colleagues, Doh operates on assumptions about English that harms his performance as a skillful writer. In fact, he plays it safe with journal articles. He managed to co-author a paper with someone who writes English natively in order to avoid the problem of poor English. Now, he is beginning to gain enough confidence in his writing to venture writing some journal articles on his own. If one looks at the conference proceedings and technical reports he has written, one can see that he, in fact, can write at native-writer level competence. If he can only complete the paper quickly enough, he will have time to get a little outside feedback on his English. It will be interesting to see how he fares in his first attempts to author an article alone.

A final matter worth comment concerns efficiency. As we noticed with Billard, efficiency is an important concern that strongly affects his writing. With Doh, the drive for efficiency is still there, but he has not yet streamlined his writing process enough yet to satisfy him. The only thing that really shows a concern for efficiency in Doh's professional practice is the Web page he and about ten others around the world have created on action semantics, Doh's specific research area. Here, Doh and his computer science colleagues store all their papers and make links with other important resources on the Net to facilitate the speed of their work. This makes finding the most current research and getting entire copies of the documents extremely convenient and time-saving. Here a team of specialists around the world work in cooperation rather than competition to advance knowledge for the computer field.

6.4 Conclusion

We have seen in Doh and Billard's responses that writing practices in the computer science profession may have certain aspects that are common to many, but that individuals employ unique strategies that depend upon the unique situation they find themselves in. In addition, we have seen that the large percent of nonnative users of English in the computer field not only suffer some disadvantages professionally, but this factor affects the evolving culture of the computer science profession as well, as it engages in the review and publication process to archive its knowledge.

In the next chapter, we conclude this present study of professional writing practices in computer science. Here, material from Chapters 3 through 6 are summarized, conclusions drawn, and future research suggested.

Chapter 7

Refinement and Application

7.1 Brief Review of Research

The goal of this research has been to investigate the writing practices situated within the professional activities of a representative sample of computer scientists and to propose models that might illustrate these practices as they operate within professional contexts. As discussed earlier, these models cannot *explain* or *describe* how the writing process works. No model is sufficient enough to accurately account for all the contextual complexities involved in writing. Human beings and the contexts within which they write are far too complex for anyone to ever explain *exactly* how and why writers select the options they do to accomplish their goals. These models can, however, serve as a theoretical construct upon which assumptions can be built and then tested against real-world experience. They can identify elements of professional practice within the computer science discipline that influence the kinds of writing that occur for the accomplishment of specific professional purposes.

In Chapter 3, a number of important features of the professional computer science writing context were identified. These were assembled in Chapter 4 into several models of professional practice to suggest what kind of writing occurs, for what specific purposes, and influenced by what important factors. Then in Chapters 5 and 6, studies of a native and nonnative writer of English were conducted to understand more of how specific writers function within the model. The responses from these professionals suggested that *efficiency* is highly valued in the computer profession and strongly influences many of its practices, including its writing. Their responses also demonstrated that nonnative users of English face unique difficulties in their work that need to be addressed in English support programs if this major segment of the profession is going to attain success in a field that employs English almost exclusively.

In this final chapter, it is appropriate that we combine the information presented thus far into a refined model of professional writing practices that accounts for all the material that has been gathered up to this point—a model that not only includes all of the features identified in Chapters 3 and 4 but also the individual native and nonnative preferences identified in Chapters 5 and 6. The model does not replace those presented in previous chapters, but merely expands and organizes the elements into an algorithmic format with potential educational applications.

7.2 The Model Refined

The following model appears in *Instructional Pseudocode*, a simple instructional language developed for this project to illustrate the logical flow of professional activities as they relate to writing in the field of computer science.[1]

[1] *Instructional Pseudocode* is roughly based on pseudocode used by computer programmers.

7.2.1 INPUT: Information Retrieval

000 START
001 QUESTION: *Do you need information?*
002 ANSWER: **if** *yes*, **go to** 004
003 ANSWER: **if** *no*, **go to** 072
004 QUESTION: *Does the information currently exist?*
005 ANSWER: **if** *yes*, **go to** 008
006 ANSWER: **if** *possibly*, **go to** 008
007 ANSWER: **if** *no*, **go to** 042
008 ACTIVITY OPTION: **contact** → informed person(s)
009 MEDIA OPTION: e-mail
010 MEDIA OPTION: telephone call
011 MEDIA OPTION: letter
012 MEDIA OPTION: face-to-face
013 ACTIVITY OPTION: **attend** → appropriate event
014 MEDIA OPTION: seminar
015 MEDIA OPTION: conference
016 MEDIA OPTION: workshop
017 MEDIA OPTION: course
018 ACTIVITY OPTION: **read** → appropriate publication
019 MEDIA OPTION: technical report
020 MEDIA OPTION: conference proceedings
021 MEDIA OPTION: article in professional periodical
022 MEDIA OPTION: electronic archival (e.g., journals, SIG homepages)
023 MEDIA OPTION: book
024 QUESTION: *Is this information personally useful?*
025 ANSWER: **if** *yes*, **go to** 027
026 ANSWER: **if** *no*, **go to** 036
027 ACTIVITY OPTION: **store actively**
028 MEDIA OPTION: save e-mail response in special file
029 MEDIA OPTION: place letter in ringed-filebook or filefolder
030 MEDIA OPTION: record conversation or presentation (cassette or video)
031 MEDIA OPTION: take notes in notebook
032 MEDIA OPTION: highlight or underline information
033 MEDIA OPTION: place labeled tabs in pages of book or journal
034 MEDIA OPTION: Xerox information and place in ringed-filebook
035 MEDIA OPTION: scan information into computer and save in special file
036 ACTIVITY OPTION: **remember passively**
037 QUESTION: *Has the quantity of information become unmanageable?*
038 ANSWER: **if** *yes*, **go to** 040
039 ANSWER: **if** *no*, **go to** 071
040 ACTIVITY OPTION: **relocate information** (for more efficient access)

041 ACTIVITY OPTION: **reorganize information** (for more efficient access)

7.2.2 INPUT: Information Generation

042 ACTIVITY OPTION: **generate research idea** (to create information)
043 QUESTION: *Will the idea generate significant results?*
044 ANSWER: **if** *yes*, **go to** 046
045 ANSWER: **if** *no*, **go to** 042
046 QUESTION: *Will there be enough interest to publish/market these results?*
047 ANSWER: **if** *yes*, **go to** 049
048 ANSWER: **if** *no*, **go to** 042
049 QUESTION: *Will the time required fit my personal career schedule?*
050 ANSWER: **if** *yes*, **go to** 052
051 ANSWER: **if** *no*, **go to** 042
052 QUESTION: *Will the time required fit other time schedules?*
053 ANSWER: **if** *yes*, **go to** 055
054 ANSWER: **if** *no*, **go to** 042
055 QUESTION: *Do I have enough expertise to carry out this project?*
056 ANSWER: **if** *yes*, **go to** 062
057 ANSWER: **if** *no*, **go to** 042 or **go to** 058
058 ACTIVITY OPTION: **recruit** → research partners
059 MEDIA OPTION: solicit assistance via electronic mailing lists
060 MEDIA OPTION: solicit assistance via professional gatherings (e.g., conferences)
061 MEDIA OPTION: solicit assistance via dissemination of personal research
062 QUESTION: *Do I have enough funds/equipment to carry out this project?*
063 ANSWER: **if** *yes*, **go to** 067
064 ANSWER: **if** *no*, **go to** 042 or **go to** 065
065 ACTIVITY OPTION: **obtain** → funds and/or equipment
066 MEDIA OPTION: proposals, application forms, and letters
067 ACTIVITY OPTION: **make** → research plan
068 ACTIVITY OPTION: **conduct** → research
069 ACTIVITY OPTION: **keep** → organized records of ideas/data
070 ACTIVITY OPTION: **generate** → information/results
071 ACTIVITY OPTION: **learn** → information (and grow in expertise)

7.2.3 OUTPUT: Information Dissemination

072 QUESTION: *Do I possess significant information?*
073 ANSWER: **if** *yes*, **go to** 075
074 ANSWER: **if** *no*, **go to** 001
075 QUESTION: *Who will be interested in and/or benefit from this information?*

076 ANSWER: **if** *the general public*, **go to** 082
077 ANSWER: **if** *students*, **go to** 100
078 ANSWER: **if** *governing personnel/administrators*, **go to** 121
079 ANSWER: **if** *the general computer science profession*, **go to** 134
080 ANSWER: **if** *specialists in the same area*, **go to** 134
081 ANSWER: **if** *specialists in related areas*, **go to** 134
082 QUESTION: *Which medium is appropriate?*
083 ANSWER: **if** *spoken*, **go to** 085
084 ANSWER: **if** *written*, **go to** 090
085 ACTIVITY OPTION: **create** → message (with textual support)
086 MEDIA OPTION: speech (supported with text or notes)
087 MEDIA OPTION: lecture (supported with text, notes, and/or OHP transparencies)
088 MEDIA OPTION: seminar (supported with notes, OHP transparencies, handouts)
089 MEDIA OPTION: workshop (supported with notes, OHP transparencies, handouts)
090 ACTIVITY OPTION: **create** → document
091 MEDIA OPTION: magazine article
092 MEDIA OPTION: brochure
093 MEDIA OPTION: instruction manual
094 MEDIA OPTION: electronic documentation
095 MEDIA OPTION: book (or part of a book)
096 ACTIVITY OPTION: **determine** → authors and acknowledgements
097 ACTIVITY OPTION: **obtain** → feedback on draft(s)
098 ACTIVITY OPTION: **revise** → document
099 ACTIVITY OPTION: **disseminate** → info to general public; GO TO 179
100 QUESTION: *Which medium is appropriate?*
101 ANSWER: **if** *spoken*, **go to** 103
102 ANSWER: **if** *written*, **go to** 108
103 ACTIVITY OPTION: **create** → message (with textual support)
104 MEDIA OPTION: speech (supported with text or notes)
105 MEDIA OPTION: lecture (supported with text, notes, and/or OHP transparencies)
106 MEDIA OPTION: seminar (supported with notes, OHP transparencies, handouts)
107 MEDIA OPTION: workshop (supported with notes, OHP transparencies, handouts)
108 ACTIVITY OPTION: **create** → document
109 MEDIA OPTION: practice exercises/problems
110 MEDIA OPTION: computerized tutorials
111 MEDIA OPTION: quiz or test
112 MEDIA OPTION: midterm and/or final exam

113 MEDIA OPTION: entrance examination (particularly in Asia)
114 MEDIA OPTION: instruction manual
115 MEDIA OPTION: electronic documentation
116 MEDIA OPTION: textbook (or part of a textbook)
117 ACTIVITY OPTION: **determine** → authors/acknowledgements (if appropriate)
118 ACTIVITY OPTION: **obtain** → feedback on draft(s)
119 ACTIVITY OPTION: **revise** → document
120 ACTIVITY OPTION: **disseminate** → info to students; **go to** 179
121 ACTIVITY OPTION: **obtain** → request for information
122 ACTIVITY OPTION: **obtain** → official forms or computer templates
123 ACTIVITY OPTION: **obtain** → submission rules/standards
124 ACTIVITY OPTION: **identify** → specific readers
125 ACTIVITY OPTION: **identify** → readers' unwritten expectations
126 ACTIVITY OPTION: **create** → document
127 MEDIA CONSIDERATION: adjust content to match reader expectations
128 MEDIA CONSIDERATION: adjust language to match reader expectations
129 MEDIA CONSIDERATION: adjust format to match reader expectations
130 ACTIVITY OPTION: **determine** → authors
131 ACTIVITY OPTION: **obtain** → feedback on draft(s)
132 ACTIVITY OPTION: **revise** → document
133 ACTIVITY OPTION: **disseminate** → info to governing personnel; **go to** 179
134 QUESTION: *Is the information HIGHLY significant?*
135 ANSWER: **if** *yes*, **target** → MOST competitive dissemination channels
136 ANSWER: **if** *no*, **target** → LESS competitive dissemination channels
137 QUESTION: *Is the information preliminary or complete?*
138 ANSWER: **if** *preliminary*, **go to** 140 (give preference to 142-152)
139 ANSWER: **if** *complete*, **go to** 140 (give preference to 153-154)
140 ACTIVITY OPTION: **gather** → research data and related publications
141 ACTIVITY OPTION: **select** → appropriate media
142 MEDIA OPTION: keynote speech (with textual support)
143 MEDIA OPTION: conference presentation (and published proceedings)
144 MEDIA OPTION: lecture (with textual support)
145 MEDIA OPTION: seminar (with textual support)
146 MEDIA OPTION: workshop (with textual support)
147 MEDIA OPTION: general correspondence
148 MEDIA OPTION: instruction manual
149 MEDIA OPTION: electronic documentation
150 MEDIA OPTION: technical report
151 MEDIA OPTION: article in SIG newsletter or SIG journal
152 MEDIA OPTION: letter from publication editor or SIG chair
152 MEDIA OPTION: review of information/product
153 MEDIA OPTION: journal article

154 MEDIA OPTION: book (or part of a book)
155 QUESTION: *Is the media primarily written or spoken?*
156 ANSWER: **if** spoken, **go to** 158
157 ANSWER: **if** written, **go to** 165
158 ACTIVITY OPTION: **create** → message (and textual support)
159 ACTIVITY OPTION: **determine** → authors/acknowledgements (if appropriate)
160 ACTIVITY OPTION: **obtain** → feedback on draft(s)
161 ACTIVITY OPTION: **revise** → text(s)
162 ACTIVITY OPTION: **disseminate** → info to audience
163 ACTIVITY OPTION: **obtain** → feedback on information
164 ACTIVITY OPTION: **establish** → new contacts; **go to** 179
165 ACTIVITY OPTION: **create** → template with section divisions
166 ACTIVITY OPTION: **paste in** → any relevant info from other documents
167 ACTIVITY OPTION: **fill in** → sections in the order of personal preference
168 ACTIVITY OPTION: **determine** → authors/acknowledgements
169 ACTIVITY OPTION: **print** → hard copy (periodically)
170 ACTIVITY OPTION: **review** → hard copies and make revisions on paper
171 ACTIVITY OPTION: **revise** → electronic document accordingly
172 ACTIVITY OPTION: **repeat** → 169-171 until satisfied
173 ACTIVITY OPTION: **send** → hard or electronic copy out for feedback
174 ACTIVITY OPTION: **revise** → electronic document accordingly
175 ACTIVITY OPTION: **send** → to *gatekeepers* (editors, referees, conf. committee)
176 **If** *rejected unconditionally*, **go to** 001
177 **If** *rejected conditionally*, **make** → requested revisions; **go to** 175
178 **If** *accepted*, **go to** 179
179 STOP or **go to** 000

7.3 Potential Educational Applications

Though the research conducted in this dissertation merely begins to lay the foundation for further, more focused, writing research in the field of computer science, there is potential for some educational applications even at this early stage of investigation. One potential application would be a general orientation to the work, and writing situated within that work, for university students just beginning their computer science studies. By exposing students to the models, the brief descriptions of genres, and the dialogs with professionals about their work and their writing, students would be better able to understand the context within which computer scientists write before beginning their own writing coursework. This kind of orientation would give weight to a writing instructor's claim that writing is important in computer science, and it would enable students to recognize some of the factors that influence the shape and content of that writing. Since affective factors, such as personal motivation, appear to strongly influence any learning endeavor, it might be possible for writing instructors to employ this material in the early stages of writing instruction to increase student interest and to demonstrate that student effort and time are directed toward a purpose they can easily envision. As many university educators testify, one of the most frequently asked questions by college students is "What does this material have to do with my major?" Good orientation early in a writing program will eliminate this question in student minds and build greater interest in writing coursework.

Another potential educational application involves making students aware of their own writing and research process. If efficiency is an important goal in the computer discipline, students could study the graphic and algorithmic models of professional writing and then develop a model of their own writing process and activities. By representing their writing activities on paper, students could better reflect on the efficiency of their own writing practices as well as obtain feedback from their peers. Developing a model of one's personal writing procedures would bring weaknesses to light that might have hitherto been unnoticed.

A third educational application involves teachers. Since the vast majority of instructors who teach writing and develop course material for students in the science and technologies appear to have little educational or employment experience in the discipline they support (Orr, 1995), the material in this dissertation could be used by teachers of computer science students to familiarize them with some genuine practices and products characteristic of writing in the computer field. It would also familiarize teachers with some of the differences in problems and writing strategies between native and nonnative writers of English. Without such knowledge, teachers would be unable to judge the usefulness of composition and technical writing textbooks for their own writing courses as well as make them unable to create effective materials of their own. By reading this dissertation, writing teachers supporting the computer sciences would gain enough of an overview of writing in CS to enable them to make better curriculum design decisions and provide them a foundation to begin additional investigations of their own.

Finally, computer scientists, themselves, would benefit from reading this research material for the purpose of comparing their own writing situation with that of their peers. Just as students in the computer science field would learn from reflecting on their own writing procedures, established professionals would likely gain even more from reflecting on their writing. By seeing the perspectives and writing strategies of others, experienced computer scientists might better understand their own habits and better adjustment their own practices to increase their personal productivity.

7.4 Future Research Applications

In this research, several models of professional writing practices in the computer science discourse community have been proposed. These models are not definitive, but simply lay the foundation for continued research. English programs that support the sciences and technologies must understand the nature of English as it is used within disciplinary discourse communities and be able to provide efficient and effective orientation for nonnative speakers who seek professional membership. Particularly in a field such as computer science, where the vast majority of professionals do not use English natively, and yet, virtually all the discipline's knowledge is disseminated and archived in English, competent English language support is crucial for the community to effectively accomplish its goals. A good English support program must be grounded in research. It must base its instructional material and methodologies on accurate information about the field it serves and about the human language-learning process. This dissertation contributes to that work.

Broadly speaking, it serves the general ESP community by exploring how English in a profession might be studied. Narrowly speaking, it serves the University of Aizu by laying further groundwork for an informed English language curriculum. The specific task of this research has been to explore a representative sample of writing practices that occur in computer science, identify potential forces that influence it, and propose a general model of how writing functions to accomplish professional goals. The research has also provided a glimpse of writing in the computer profession from both a native and nonnative English writer's perspective. The research that must follow the work accomplished here is plentiful.

One task that must be carried out is the continual collection of data to test these models against real-world experience. Where do the models describe practices inaccurately? Where should detail be added to capture more of the complexity that is involved in producing texts? What other forces exist that influence the writing process? What practices will change as technology changes? These are a few of the questions that need to be addressed.

Other topics of study must include deeper analysis of the texts themselves. What features are conventional? How much freedom is permitted by the *gatekeepers* in material submitted to journals and conferences? How do textual expectations change from periodical to periodical and from conference to conference? Why do these features occur and how did they develop? These are other questions that need to be answered.

It is also important that *efficiency* be considered in future research. Which research and writing practices are inefficient and which ones are efficient? Does efficiency naturally improve with age and length of experience? Is efficiency affected by personality or nationality or gender? Should efficiency be included in the language curriculum? Answers to these questions would be welcomed by many in the computer profession.

And finally, the topic of language education must be investigated, for that is the goal of this work. What kind of material is worthy of attention in the English language classroom? Authentic samples of texts and speech? What features of English need to be addressed for particular kinds of learners? How can context be taught? Which language skills can be taught apart from context and which ones cannot? Would team teaching efforts between computer science faculty and language instructors improve language learning?

All these, and many more questions still await those who are charting new educational territory in ESP. As more and more universities move to discipline-specific language curriculums, the need for this kind of research will continue to grow. Language researchers and educators still know far too little about how humans learn languages and employ them to meet their needs. The research begun here must continue.

References

Adams Smith, D. E. (1984). Medical discourse: Aspects of author's comment. *The ESP Journal, 3*, 25-36.

Alexander, P. A. & Judy, J. E. (1989). The interaction of domain-specific and strategic knowledge in academic performance. *Review of Educational Research, 58*, 375-404.

1993 Annual review: School of Computer science and engineering. Aizuwakamatsu, Japan: University of Aizu.

Ard, J. (1983). The role of the author in scientific discourse. Paper presented at the annual American Applied Linguistics Meeting, Minneapolis, MN, December, 1983.

Barber, C. L. (1962). Some measurable characteristics of modern scientific prose. In *Contributions to English syntax and phonology*: 1-23. Stockholm: Almquist & Wiksell.

—. (1962). Some measurable characteristics of modern scientific prose. In Swales (ed.) (1985). *Episodes in ESP: A source and reference book on the development of English for science and technology.* Oxford: Pergamon Press.

Bavelas, J. B. (1978). The social psychology of citations. *Canadian Psychological Review, 19*, 158-63.

Bazerman, C. (1984). Modern evolution of the experimental report in physics: spectroscopic articles in *Physical Review*, 1893-1980. *Social Studies in Science, 14*, 163-96.

—. (1985). Physicists reading physics: Schema-laden purposes and purpose laden schema. *Written Communication, 2*, 3-24.

Bereiter, C. & Scardamalia, M. (1987). *The psychology of written composition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Berkenkotter, C. & Huckin, T. H. (1995). *Genre knowledge in disciplinary communication: Cognition, culture, and power.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Bhatia, V. K. (1993). *Analysis genre: Language use in professional settings.* London: Longman.

Billard, E. A. (1994). Instabilities in learning automata playing games with delayed information. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1160-1165). Los Alamitos, CA: IEEE Computer Society Press.

—. (1994). Learning in multi-level stochastic games with delayed information. *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence.* (pp. 86-93). San Francisco, CA: Morgan Kaufmann.

—. (1994). *Learning in single- and multi-level games with delayed information* (Tech. Rep. No. 94-1-011). Aizuwakamatsu, Japan: University of Aizu.

—. (in press). Stability of adaptive search in multi-level games under delayed information. *IEEE Transactions on Systems, Man, and Cybernetics.*

Blau, J. (1974). Patterns of communication among theoretical high energy physicists. *Sociometry, 37*, 391-406.

Bradford, D. (1984). The persona in microcomputer documentation. *IEEE Transactions on Professional Communication, 27*, 65-68.

Bramki, D., & Williams, R. (1984). Lexical familiarization in economics text, and its pedagogic implications for reading comprehension. *Reading in a Foreign Language, 2*(1), 169-181.

Brookshear, J. G. (1994). *Computer science: An overview.* (4th ed.). Redwood City, CA: Benjamin/Cummings.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. Educational Researcher, 18, 32-42.

Browning, C. (1984). *Guide to effective software technical writing.* Englewood Cliffs, NJ: Prentice Hall.

Bruce, N. (1984). Rhetorical constraints and information structure in medical research report writing. *English for Medical and Paramedical Purposes Newsletter, 1*(1), 5-18.

Capron, H. L. (1992). *Essentials of computing.* Redwood City, CA: Benjamin/Cummings.

Chandani, A. (1985). Writing styles of abstracts in occupational therapy journals. *British Journal of Occupational Therapy, 48*, 244-246.

Cooper, C. (1985). Aspects of article introductions in IEEE publications. Unpublished doctoral dissertation, University of Aston, Birmingham, England.

Darian, S. (1982). The role of definitions in scientific and technical writing: Forms, functions and properties. In J. Hoedt et al. *Pragmatics and LSP.* Copenhagen: Copenhagen School of Economics.

Draskau, J. (1986). A valency-type analysis of four monolexical verbs and two polylexical verbs in English LSP texts from two subject fields, marine engineering and veterinary science. *UNESCO ALSED-LSP Newsletter, 9*(2), 25-42.

Dubois, B. L. (1980). The use of slides in biomedical speeches. *English for Specific Purposes, 1*, 45-50.

—. (1982). The construction of noun phrases in biomedical journal articles. In J. Hoedt et al. *Pragmatics and LSP.* (pp. 49-67). Copenhagen: Copenhagen School of Economics.

—. (1988). Citation in biomedical journal articles. *English for Specific Purposes, 7*, 181-94.

Een, J. A. (1982). Tense usage in reporting research in geotechnical writing. In *Working Papers in ESL(2).* (pp. 72-91). Minneapolis: University of Minnesota.

Ewer, (1979). *The modals in formal scientific discourse: function, meaning and use.* Santiago, Chile: Research Report, Department of English.

Flower, L. (1985). *Problem-solving strategies for writing.* (2nd ed.). San Diego: Harcourt Brace Jovanovich.

Gunawardena, C. N. (1989). The present perfect in the rhetorical divisions of biology and biochemistry journal articles. *English for Specific Purposes, 8*, 265-274.

Goetz, P. W. (Ed.). *The New Encyclopedia Britannica.* (15th ed., Vol. 1). Chicago:

Encyclopedia Britannica, Inc.

Hall, D., Hawkey, R., Kenny, B, & Storer, G. (1986). Patterns of thought in scientific writing: A course in information structuring for engineering students. *English for Specific Purposes, 5*, 147-160.

Hayes, J. R. & Flower, L. (1980). Identifying the organization of writing processes. in L. W. Gregg & E. R. Steinberg (Eds.). *Cognitive processes in writing.* (pp. 3-30). Hillsdale, NJ: Lawrence Erlbaum Associates.

Heslot, J. (1982). Tense and other indexical markers in the typology of scientific texts in English. In J. Hoedt et al. *Pragmatics and LSP.* (pp. 83-103). Copenhagen: Copenhagen School of Economics.

Holes, C. (1984). Textual approximation in the teaching of academic writing to Arab students: A contrastive approach. In Swales and Mustafa (eds.) *English for specific purposes in the Arab world.* (pp. 228-42) Birmingham, UK: The University of Aston.

Horowitz, D. M. (1986). What professors actually require: academic tasks for the ESL classroom. *TESOL Quarterly, 20*, 445-62.

Horsella, M., & Perez, F. (1991). Nominal compounds in chemical English literature: Toward an approach to text typology. *English for Specific Purposes, 10*, 125-138.

Hutchinson, T., & Waters, A. (1987). *English for specific purposes: A learning-centered approach.* Cambridge: Cambridge University Press.

Inman, M. (1978). Lexical analysis of scientific and technical prose. In Trimble M. T., Trimble, L. & Drobnic, K. (Eds.) (1978). *English for specific purposes: Science* *and technology.* Corvallis: Oregon State University.

Jacobson, W. H. (1986). An assessment of the communication needs of non-native speakers of English in an undergraduate physics lab. *English for Specific Purposes, 5*, 173-188.

Johns, A. M. & Dudley-Evans, T. (1991). English for specific purposes: International in scope, specific in purpose. *TESOL Quarterly, 25*, 297-314.

Kinkoph, S., Fulton, J., & Oliver, K. (1994). *Computers: A visual encyclopedia.* Indianapolis, IN: Alpha Books.

Lackstrom, J., Selinker, L., & Trimble, L. (1973). Technical rhetorical principles and grammatical choice. *TESOL Quarterly, 7*, 127-36.

Lackstrom, J. (1978). Teaching modals in EST discourse. In Trimble, Trimble and Drobnic (Eds.) *English for specific purposes: Science and technology.* Corvallis: Oregon State University, 53-73.

Lauer, J. M. & Asher, J. W. (1988). *Composition research: Empirical designs.* New York: Oxford University Press.

Lenze, J. (1988). Course descriptions across the disciplines. *Papers in Applied Linguistics - Michigan, 3*(2), 53-72.

Love, A. (1993). Lexico-grammatical features of geology textbooks: Process and product revisited. *English for Specific Purposes, 12*, 197-218.

Mackay, R., & Mountford, A. (Eds.). (1978). *English for specific purposes: A case study approach.* London: Longman.

Malcom, L. (1987). What rules govern tense usage in scientific articles? *English for*

Specific Purposes, 6, 31-44.

McKenna, E. (1987). Preparing foreign students to enter discourse communities in the U.S. *English for Specific Purposes, 6*, 187-202.

Meyers, G. (1985). Texts as knowledge claims: The social construction of two biologists' articles. *Social Studies of Science, 15*, 593-630.

North, S. M. (1987). *The making of knowledge in composition: Portrait of an emerging field.* Portsmouth, NH: Heinemann.

Nunnally, J. C. (1978). *Introduction to psychological measurement.* New York: McGraw-Hill.

Orr, T. (1995). Studying the professions we serve. *TESOL ESP News, 4*(2): 11-12.

Oster, S. (1981). The use of tenses in reporting past literature. In L. Selinker, E. Tarone, and V. Hanzeli (Eds.) *English for Academic and Technical Purposes: Studies in Honor of Louis Trimble.* (pp. 76-90). Rowley, MA: Newberg Press.

Peng, J. (1987). Organizational features in chemical engineering research articles. *ELT Journal, 1*, 79-116.

Pettinari, C. (1982). The function of a governmental alternative in fourteen surgical reports. *Applied Linguistics, 4*, 55-76.

Popken, R. L. (1987). A study of topic sentence use in academic writing. *Written Communication, 4*, 209-28.

Porter, J. E. (1992). *Audience and rhetoric.* Englewood Cliffs, NJ: Prentice Hall.

Ramani, E., Chacko, T., Singh, S. J., & Glendenning, E. H. (1988). An ethnographic approach to syllabus design: A case study of the Indian Institute of Science, Bangalore. *English for Specific Purposes, 7*, 81-90.

Robinson, P. (1980). *ESP (English for specific purposes): The present position.*. Oxford: Pergamon Press.

—. (1991). *ESP today: A practitioner's guide.* Hertfordshire, UK: Prentice Hall.

Rogoff, B. & Lave, J. (Eds.). (1984). *Everyday cognition: Its development in social context.* Cambridge, MA: Harvard University Press.

Rounds, P. L. (1982). Hedging in written academic discourse: Precision and flexibility (mimeo). Ann Arbor: The University of Michigan.

Salager, F. (1983). The lexis of fundamental medical English: Classificatory framework and rhetorical function (a statistical approach). *Reading in a Foreign Language, 1*, 54-64.

Shriver, K. A. (1992). Connecting cognition and context in composition. In G. Kirsch & P. A. Sullivan (Eds.). *Methods and methodology in composition research.* Carbondale, IL: Southern Illinois University Press.

Silva, T. (1990). Second language composition instruction: Developments, issues, and directions in ESL. In B. Kroll (Ed.). *Second language writing: Research insights for the classroom.* Cambridge: Cambridge University Press.

Spencer, D. (Ed.). (1992). *Webster's New World Dictionary of Computer Terms.* (4th ed.). New York: Prentice Hall.

Stevenson, J. L., & Sprachman, S. (1981). *Reading the social sciences in English.* London: Longman.

Strevens, P. (1988). ESP after twenty years: A re-appraisal. In M. Tickoo (Ed.) *ESP: State of the art* (pp. 1-13). Singapore: SEAMEO Regional Language Centre.

Swales (Ed.) (1985). *Episodes in ESP: A source and reference book on the development of English for science and technology.* Oxford: Pergamon Press.

—. Genre analysis: English in academic and research settings. Cambridge: Cambridge University Press.

Swales, J. & Najjar, S. (1987). The writing of research article introductions. *Written Communication, 4*, 175-92.

Swales, J. & Feak, C. (1994). *Academic writing for graduate students: A course for nonnative speakers of English.* Ann Arbor, MI: University of Michigan Press.

Tarone, E., Dwyer, S, Gillette, S. & Icke, V. (1981). On the use of the passive in two astrophysics journal papers. *The ESP Journal, 1*(2), 123-40.

Toffler, A. (1970). *Future shock.* New York: Bantam Books.

—. (1980). *The third wave.* New York: Bantam Books.

—. (1983). *Previews & premises.* New York: Bantam Books.

Trimble, L. (1985). *English for science and technology: A discourse approach.* Cambridge: Cambridge University Press.

Weissberg, R. (1984). Given and new: Paragraph development models for scientific English. *TESOL Quarterly, 18*, 485-500.

West, G. K. (1980). That-nominal constructions in traditional rhetorical divisions of scientific research papers. *TESOL Quarterly, 14*, 483-9.

Wingard, P. (1981). Some verbs forms and functions in six medical texts. In L. Selinker, E. Tarone, and V. Hanzeli (Eds.) *English for Academic and Technical Purposes: Studies in Honor of Louis Trimble.* (pp. 53-64). Rowley, MA: Newberg Press.

Wittrock, M. C. (1985, April). *Cognitive processes in the learning and teaching of science.* Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Yanoff, K. L. (1988). The rhetoric of medical discourse: An analysis of the major genres. *Dissertation Abstracts International, 49*, 3011.

Appendix A

Survey Questions

The questions listed below have been taken from six different surveys and numerous private interviews with the University of Aizu computer science faculty over a research period of two year and one-half years. Only those questions that have relevance to this particular research project have been listed.

A.1 Questions about Reading Practices

1. How many pages (hardcopy) of English do you typically read each year?
2. How many pages (hardcopy) of Japanese do you typically read each year?
3. How many pages (hardcopy) of other languages do you typically read each year?
4. How many hours do you typically read English on your computer screen each year?
5. How many hours do you typically read Japanese on your computer screen each year?

6. How many hours do you typically read other languages on your computer screen each year?

A.2 Questions about Writing Practices

1. What is your area of specialization?

2. What type of work-related documents (in English) does your specific work in computer science require you to produce?

3. How many years have you been writing such documents?

4. How did you learn to write these kinds of documents?

   Course(s)?

   Advice from graduate school professors?

   Advice from research supervisors?

   Advice from research team members?

   Advice from your dissertation advisors?

   Advice from editors?

   Advice from books? (Titles?)

   Imitate other documents?

   Other ways? (Explain)

5. Approximately, how many pages of English do you write each year on average?

6. What kind of work-related documents do you find most difficult to write?

7. What other languages do you write in professionally?

8. Do the style, format, organization, logic, (any characteristics) of the documents you write in English differ from the same kind of documents you write in your native language?

9. Through which professional organizations and publications would you most like to disseminate you knowledge/opinions?

A.3 Questions to Accompany a Sample Document

1. Document Type:

2. Number Written in a Typical Year:

3. Number Published in a Typical Year:

4. Subject Matter:

5. Intended Readers:

6. Purpose of Document:

7. Typical Length:

8. Typical Number of Citations:

9. Typical Number of Formulas:

10. Typical Number of Graphs, Diagrams, Illustrations:

11. Language of Final Product:

12. Language of Drafts:

13. Any Outside Writing Assistance:

14. How did you learn to write this kind of document?

15. Should our students learn to write this kind of document?

16. If yes, from who?

A.4 Questions Related to Plans for English Usage in Computer Science Courses

1. Name of course:

2. Name of English textbooks you plan on using:

3. Specific chapters or pages you will assign as reading material:

4. Other English reading material planned (OHPs, handouts, white-board notes, etc.):

5. English writing assignments you plan to give:

6. Kinds of English documents will you require:

7. Types of English quizzes, tests, or examinations you plan to give:

8. Percent of course devoted to lectures in English:

A.5 Additional Questions Frequently Asked in Brief Personal Interviews

1. Describe the professional computer science organizations to which you belong?

   How does one becomes a member?

   What kind of people usually join?

What kind of publications does the organization put out?

Describe the submission and review process?

What are the conferences like?

2. How does one professional computer science organization (membership, publications, conferences, etc.) differ from another? (e.g. ACM and IEEE)

3. How do professional computer science publications in your country differ from international ones such as those of the ACM or IEEE?

4. Have you ever served as an editor or referee?

5. How did you evaluate papers?

6. What kind of advice have you often received from editors or reviewers?

Appendix B

Referee Reports

The following criteria are frequently used by referees in computer science to evaluate potential conference papers and journal articles. Since there is frequent redundancy, and because *some* referee report formats are confidential, the contents of these reports have been organized and listed below without reference to the specific journals or conferences from which the information was obtained. The only exception to this is the first section on Referee Ethics which states public professional policy. It should noted, however, that local computer scientists confirmed that the anonymous criteria listed below for conferences and journals are typically used by "gatekeepers" in the field to judge a paper's qualifications for entrance to the most competitive avenues of dissemination in computer science.

B.1 Referee Ethics and Guidelines

B.1.1 Ethics Statement

The following resolution was passed by the Transactions Advisory Committee of the IEEE[1] Computer Society in June 1991.

In the interest of fostering excellence in research, the IEEE Computer Society recognizes that the peer review process plays a central role in research. When it works well, the review process helps improve the quality and accuracy of submissions. When it breaks down, the review process greatly delays the publication of worthy material. This statement of referee ethics is intended to assure that the review process works most effectively. By following the ethics here, each referee should receive personal satisfaction from a contribution to the profession for each review produced. Moreover, when the referee authors a paper, the referee will receive the benefits of an effective and efficient peer review process.

Professional Responsibility of Referees

- Each researcher should referee on an annual basis at least three times the number of papers submitted by that researcher.
- A referee should complete the refereeing of a paper by the assigned deadline, or ask a qualified colleague to review, or should return the unreviewed paper immediately. If a referee returns a paper without review, it would be helpful for the referee to recommend alternative reviewers.
- The referee must respect the privileged content of the submitted material.

[1]The Institute of Electrical and Electronics Engineers, Inc.

- If a referee recommends acceptance of an article, the referee is assuring the accuracy of the technical content, originality, and the proper credit to previous work to the best of the referee's ability to judge these aspects. The referee can seek the assistance of a colleague if this is helpful.

B.1.2 Guidelines for Reviewers

EXAMPLE ONE

Reviewers should see themselves as protectors of the quality of the *Transactions*[2] as well as of the reputation of the authors who submit papers. It is the Reviewer's responsibility to make sure only high quality papers are published, and that the Author(s) is protected from putting poor work into print. From this perspective, the Reviewer should not only read the papers thoroughly to find flaws, but should also make recommendations to the author(s) as to how the paper might be improved.

Types of Reviews

Reviews should be divided into three parts:

(a) a technical review, pointing out technical errors and making recommendations for improvements in the paper;

(b) an editorial review, pointing out the relevance, importance, and originality of the work; and

(c) a review of the presentation style and English.

[2]The IEEE is organized into nearly 40 professional *societies*, each of which publishes one or more technical publications known as *transactions*. The *Transactions* are channels for dissemination of the profession's most significant research. The ACM (Associate for Computing Machinery) is the other major professional computer science organization. Among its more than 50 professional periodicals, the *ACM Transactions* series is also recognized as the computer profession's top journals.

Conflicts of Interest

Before reviewing a paper, the Reviewer should make sure that there is no conflict of interest in his/her reviewing the paper. Examples of cases which could cause conflict of interest include:

(a) Papers by an author with which the Review has coauthored a paper recently.

(b) Papers by an author at the same department of in a closely related discipline of the same university as the Reviewer.

Papers by an author who was a recent student or thesis advisor of the Reviewer.

If the Reviewer feels that his or her decision will be affected, he or she should return the paper to the Editor with a statement explaining the conflict of interest.

EXAMPLE TWO

Instructions to Referees: For a contribution to be acceptable for publication in *Transactions on...* either as PAPER, CONCISE PAPER, or as a CORRESPONDENCE, it must comprise novel material not previously published in a technical journal. The novelty will usually lie in original results, methods, observations, concepts, or applications, but may also reside in syntheses of or new insights into previously reported research. In a regular PAPER (35 pages or less), the title, abstract, introduction, and summary should be sufficiently informative to make the contributions of the paper clear to the broadest possible audience, and to place them in context with related work. A CONCISE PAPER (12 pages or less) presents results that are important and original and are presented in a concise form. A CORRESPONDENCE (4 pages or less) is used to convey only a few principal ideas or to comment on previous work published in *Transactions on....*[3]

In addition to these fundamental requirements, acceptance for publication depends on a number of other important criteria relating to reader interest, technical content, and presentation...etc.

EXAMPLE THREE

Note to Reviewers: If major revisions are recommended, you should point these out as specifically as possible. Please differentiate optional cosmetic changes from those you consider essential. If the revisions required are extensive, it is perhaps best the reject the paper and recommend preparation of a new, largely revised manuscript for resubmission. If you judge the manuscript inappropriate on the basis of reader interest, please suggest a more appropriate journal to the author(s). Manuscripts with little or no salvageable material should be rejected outright and later resubmission discouraged.

B.2 Conference Papers

B.2.1 Evaluation for the Program Committee

EXAMPLE ONE

Directions: Evaluation the quality of the paper on a scale of 1 (very low) to 10 (very high).

- Scientific Quality ___. Reason: _______.

[3] Other *Transactions* only have PAPERS and CORRESPONDENCES. The PAPERS are usually 35 pages or less and the CORRESPONDENCES are 15 pages or less.

- Originality ___. Reason: _______
- Significance of Contribution ___. Reason: _______
- Clarity & Readability ___. Reason: _______
- Relevance to Conference ___. Reason: _______

Overall Recommendation (1-10): ____.

(Reject 1-3, Weak Reject 4-5, Weak Accept 6-7, Accept 8-10)

Reviewer's Familiarity with the Topic: Low___ Medium___ High___

Significant Problems Addressed: __________...

Comment on References: __________...

Comment on Ideas: __________...

Comment on Results: __________...

Will the paper require much rewriting if accepted? YES or NO

If YES, where? __________...

EXAMPLE TWO

Quality Scoring Scale: 1(low)-7(high)

Quality Criteria:

- Conceptual/Foundational Contribution (New approach, models, problems?)
- Technical Development (New techniques?)
- Social Interest (Broad? Narrow?)

- Open Problems Settled
- Transferable Implementation and Application

Confidence Scoring Scale:

- 1 (I didn't actually look at it.)
- 2 (I made a rough guess.)
- 3 (I'm pretty sure about my decision.)
- 4 (I'm sure about my decision.)

B.2.2 Feedback for Authors

EXAMPLE ONE

Detailed comments and recommendations for the authors, including how to revise the paper: ____________...

EXAMPLE TWO

Ranking: Please evaluate the items listed below according to the following scale.

+3: Strong Accept (It's as good as any top paper in good conferences.)

+2: Accept (It's comparable to good papers in good conferences.)

+1: Weak Accept (I vote for acceptance but won't argue for it.)

0: Neutral (I don't like it, but I won't object if other like it.)

-1: Weak Reject (I would rather see the paper not accepted.)

-2: Reject (I will argue to reject this paper.)

-3: Strong Reject (In the lowest 10% of papers reviewed.)

Originality: ______...

Significance: ______...

Correctness: ______...

Presentation: ______...

Relevance to Conference: ______...

OVERALL RATING: ______...

Amount of Revision Required: LARGE, MODERATE, SMALL

Referee's Confidence Rating: LOW, AVERAGE, HIGH

Give a short summary of the rationale for your decisions (Maximum 3 lines): __________...

Any detailed comments to authors: __________...

B.3 Journals

B.3.1 Evaluation for Journal Editors

EXAMPLE ONE

Quality of Paper:

Award Quality___, Excellent___, Good___, Fair___, Poor ___

Recommendation to Editors:

___Accept with minor changes as a___Regular Paper___Technical Correspondence.

__Prepare a major revision for re-review as a__Regular Paper__Technical Correspondence.

__Reject.

If the paper is rejected for publication, the authors should:

__Prepare a major revision and resubmit to [Journal Name] as a "new" paper.

__Submit to another publication, such as:_____.

__ Regard the paper as not publishable.

B.3.2 Feedback for Authors

EXAMPLE ONE

1. Does the introduction state the purpose of the paper?

2. Is the significance of the paper, relative to the existing literature, explained?

3. Is the paper clearly organized?

4. Are there adequate references to other research?

5. Is the paper cogent?

6. Does the author explain well *what* was done?

7. Does the author explain well *why* it was done?

8. Is the paper appropriate in scope for [Journal Name]?

9. Is the paper well organized?

10. Relative to its content and scope, is the length of the paper appropriate?

11. Is the English satisfactory?

12. How readable is the paper for an educated reader who is not a detail specialist in the particular field?

13. Disregarding the technical content in the paper, how would you rate the quality of the presentation?

14. Is the paper of current interest to a reasonable segment of the [Journal Name] readership?

15. Relative to the current level of reader interest in the paper, how is this interest likely to change during the next five years?

16. Within its particular field of specialization, is the paper subject matter important?

17. What percentage of the current entire [Journal Name] readership do you estimate will read this paper? __% will benefit from this paper? __%

Note to Referee: Please make very detailed technical and editorial comments and suggestions directly on the manuscript....Particular attention should be given to details that guide possible revisions, or that clear explain your reasons for rejection.

EXAMPLE TWO

A. Reader Interest

1. Is the paper of current interest to a reasonable segment of the [Journal Name] Readership? YES – PERHAPS – NO

2. To what extent is material in the paper likely to be used by other researchers and practitioners? LARGE - AVERAGE - SMALL

B. Content

1. is the paper technically sound? YES - PARTIALLY - NO - WAS NOT ABLE TO CHECK IN DETAIL

2. Is the bibliography adequate? YES - NO - SEE RECOMMENDED ADDITIONS AND/OR DELETIONS

B.1

If part of this paper's stated contribution is an algorithm and/or its implementation, then please complete this section.

1. Is the implementation being made available to other researchers? YES - NO

2. If not, is sufficient information provided for the implementation to be reproduced? YES - NO

3. Have results been shown for a sufficient number of real and standard/appropriate data sets in order to demonstrate the potential utility of the algorithm? YES - NO - SEE SPECIFIC SUGGESTIONS

4. Are the "breaking points" or "failure modes" of the implemented algorithm documented in the paper? YES - PARTIALLY - NO

5. Does this paper adequately compare the proposed algorithm against existing techniques and demonstrate it superior performance (based on accuracy, speed, etc.)? YES - MAYBE - NO - NOT APPLICABLE

B.2

If part of this paper's stated contribution is a new theory or concept, then please complete this section.

1. Does this paper attempt to formalize a (sub)problem that has not previously been considered? YES - NO

2. If the problem has been considered before, is it being addressed here in a new way that is new and interesting in some regard? YES..NO

3. Are the basic assumptions on which the theory of conceptual framework is built made clear and explicit? YES - NO

4. Do these basic assumptions reasonably approximate or generalize to the real-world characteristics of the intended applications? YES..NO..MAYBE

5. Is the intended application well motivated and well described? YES..NO

C. Presentation

1. Is the abstract an appropriate and adequate digest of the work presented? YES - NO

2. How would you rate the overall organization of the paper? SATISFACTORY - NEEDS IMPROVEMENT

3. Relative to its technical content, is the length of the paper appropriate? YES - NO (IT SHOULD BE SHORTENED)

4. Is the English satisfactory? YES - NO

Conclusions

1. Overall, does the paper offer some specific useful conclusions? YES - NO

2. Overall, are the conclusions properly supported by the results contained in the manuscript? YES - NO

EXAMPLE THREE

A. Reader Interest

(1) Is the paper of current interest to a reasonable segment of the [journal name] readership? YES - PERHAPS - NO

(2) Relative to the current level of reader interest in the paper, how is this interest likely to change during the next five years? GROWING INTEREST - RELATIVELY LITTLE CHANGE - DIMINISHING INTEREST

(3) Within this particular field of specialization (as defined, for example, by the scope of a Computer Society Technical Committee), is the topic of the paper considered important? YES, DEFINITELY - MODERATELY SO - NOT REALLY

B. Content

(1) Is the paper technically sound? YES - APPEARS TO BE, BUT I DIDN'T CHECK COMPLETELY - ONLY PARTIALLY - NO

(2) How would you describe the technical depth of the paper? EXPERT LEVEL - APPROPRIATE FOR SOMEONE WORKING IN THE FIELD - SUITABLE FOR THE NONSPECIALIST - SUPERFICIAL

(3) Does the paper make a tangible contribution to the state-of-the-art in its field? YES, DEFINITELY - TO A LIMITED EXTENT - NO

(4) Is the bibliography adequate? YES - YES, WITH CERTAIN CHANGES THAT I HAVE RECOMMENDED - NO

(5) What is your technical evaluation of the paper? AWARD QUALITY - ACCEPTABLE - MARGINALLY ACCEPTABLE - MARGINALLY UNACCEPTABLE - UNACCEPTABLE

C. Presentation

(1) Is the abstract an appropriate and adequate digest of the work presented? YES - NO

(2) Does the introduction clearly state the background and motivation in terms understandable to the non-specialist? YES - PROBABLY - NO

(3) How would you rate the overall organization of the paper? SATISFACTORY - COULD BE IMPROVED - POOR

(4) Relative to its technical content, it the length of the paper appropriate? YES - NO, SHOULD BE LENGTHENED - NO, SHOULD BE SHORTENED

(5) Is the English satisfactory? YES - NO

(6) How readable is the paper for a computer scientist or engineer who is not a specialist in this particular field? READABLE WITH ORDINARY EFFORT - PAPER IS SELF-CONTAINED, BUT A CONSIDERABLE EFFORT IS REQUIRED - IF THE DEFINITIONS OF CERTAIN CONCEPTS, TERMS, AND SYMBOLS WERE INCLUDED (NOTED BY 'DEFINE' IN THE MARGINS), READABILITY WOULD BE IMPROVED - LESS THAN HALF OF THE PAPER IS READABLE - UNREADABLE

(7) Disregarding technical content, how would you rate the quality of the presentation? AWARD QUALITY - ACCEPTABLE - MARGINALLY ACCEPTABLE - MARGINALLY UNACCEPTABLE - UNACCEPTABLE

Appendix C

Interview Questions for NS

The following questions were written to guide the initial extended interviews with the subject, Ted Billard, for the native speaker case study. These were sent to the subject a few days prior to the interviews so that there would be adequate time to think about answers and prepare illustrative material.

```
In preparation for the interviews, it would be helpful to have
the following data on each publication you have done in the past
12-18 months.  You can explain/expand this information when we
chat.

I. Projects/Results/Etc. that were disseminated via ONLY one
   public version

Title:
Authors:
Publication:
Date:
```

Intended Audience:

Purpose:

II. Projects/Results/Etc. that were disseminated in several public versions (e.g., TR to Proceedings to Article)

Title of 1st Version:

Authors:

Publication:

Date:

Intended Audience:

Communicative Function/Purpose:

Title of 2nd Version:

Authors:

Publication:

Date:

Intended Audience:

Communicative Function/Purpose:

Major Changes from 1st Version:

Title of 3rd Version:

Authors:

Publication:

Date:

Intended Audience:

Communicative Function/Purpose:

Major Changes from 2nd Version:

Here are some of the questions I may ask just in case you would like time to think about them in advance.

Concerning the field...

1. How would you describe/define the field of computer science to someone outside the field?

2. How would you describe/define a computer scientist?

3. Would you classify yourself as a computer scientist?

4. How would you describe your area of specialization?

Concerning Writing in the field...

1. How important is writing to your work?

2. What kinds of writing do you do?

   a. define/describe each type

   b. purpose of each type/audience for each type

3. Which kind of document do you write most frequently?

4. How did you learn to do this kind of writing?

Concerning Specific Genres in Computer Science...

1. Explain a particular document or series of documents that typify what you do.

   a. communicative purpose of document

   b. intended audience

   c. conventions in CS that you MUST conform to in the document

   d. writing freedoms you may exercise in this document

   e. your writing process of this document

   f. success in fulfilling your purpose

Appendix D

Articles Referenced in Case Studies

Instabilities in Learning Automata Playing Games with Delayed Information

Edward A. Billard
Faculty of Computer Science and Engineering
University of Aizu, 965-80 Japan
email: billard@u-aizu.ac.jp

Abstract **Previous results in learning automata playing sequential stochastic games showed that, with the proper choice of parameters, the players learn the optimal mixed strategy. The model is extended to games with delayed information concerning the other player's mixed strategy, as might result from the latencies in a distributed system. Predicted oscillations from a nonlinear delay differential equation and from simulations are presented. An outline of a linear stability analysis shows that as parameters are chosen to more closely approximate the optimal strategies, the system is more susceptible to delay-initiated instabilities.**

I. Introduction

We consider two players in sequential games with a high degree of uncertainty as an example of distributed decision making and adaptive systems. This work is a direct consequence of previous work in stochastic games with incomplete information [1] [2]. The previous model presented two players which update their mixed strategies according to learning automata algorithms. Each player is unaware of the pure strategy employed by the other player and, instead of a game matrix with deterministic payoffs, probabilities determine a unit gain or loss. It has been shown that, with the proper choice of parameters, the players learn the optimal strategy [1]. We extend the model to games with delayed information concerning the other player's mixed strategy, as might result from the latencies in a distributed (computing) system.

Learning automata have been successfully employed in a variety of applications, examples are communication networks [3] and the distributed processing of jobs in queueing systems [4] [5]. Games are a simple model to capture the utility of agents' decisions and games with incomplete information capture the difficulty in the decision-making process. We are particularly interested in information that may be complete, albeit delayed, and have previously examined learning automata in nonzero-sum games with delayed information where agents must decide to work together in a coalition or work alone [6]. With sufficient delays, the agents decide to work alone, a suboptimal equilibrium. In another study involving queueing systems, agents using learning automata improve performance by forming smaller groups to counteract the effects of delayed communication [7].

Delay differential equations permit the analytic solution to quite complex interactions [8]. Computational ecosystems model a large number of agents sharing resources with delayed information of other agents' strategies [9] [10]. The dynamics are described by linear delay differential equations which exhibit instabilities (including oscillations and chaos) based on the amount of delay in the system. It has also been shown that adaptive strategies, at least with instantaneous rewards, can be used to control the instabilities [9].

In Section II, we restate the problem from Lakshmivarahan and Narendra [1] and we modify the differential equation to incorporate delays. Section III shows characteristic behaviors for various parameter settings. In Section IV, we outline a proof of the conditions for instabilities and show that a good choice of parameter values, to approximate the optimal equilibrium, is a poor choice with respect to preventing delay-initiated instabilities. This has implications for the performance of learning automata in distributed environments, especially those with significant delays. Our conclusions are presented in Section V.

II. The Model with Delayed Information

Stochastic games [1] [2] model uncertainty in payoff and, as such, represent potential applications in uncertain environments. The players are modeled as learning automata which apply mixed, hence uncertain, strategies. The payoffs in a game determine the rewards or penalties and the associated dynamical behavior of the learning automata. These aspects have a well-founded background [2]; the contribution here is to incorporate delays in a player's knowledge of the other player's mixed strategy.

Manuscript received July 1, 1994.

Step 1: One Automaton - Two Strategies

Let $p(t)$ and $\bar{p}(t)$ be the probability of selecting strategy 1 and strategy 2, respectively, at time t. The probability is incremented or decremented for the next time step by

$$\Delta p = \theta \cdot \begin{cases} +\beta\bar{p} & \text{if reward on strategy 1} \\ -\beta p & \text{if reward on strategy 2} \\ -\alpha p & \text{if penalty on strategy 1} \\ +\alpha\bar{p} & \text{if penalty on strategy 2} \end{cases} . \quad (1)$$

The extent of the incremental change in the mixed strategy is determined by the three constants: β is the reward parameter, α is the penalty parameter, and θ is the step size parameter. It is assumed that $0 < \alpha < \beta < 1$ and $0 < \theta \leq 1$. Although θ can be incorporated into α and β, it is convenient to extract this term for simulation and analysis results.

Step 2: Two Players - Two Strategies

We define two players $k, l \in \{1,2\}$ in a game $\mathbf{D} = (\mathbf{D}^1, \mathbf{D}^2)$, where $\mathbf{D}^k$ respresents a stochastic payoff matrix for player k [1] [2]. Each player chooses a strategy $i, j \in \{1,2\}$, respectively, and the game is played in stages with element d_{ij}^k of $\mathbf{D}^k$ being the probability of a unit gain for player k based upon the strategy pair (i, j). With probability 1-d_{ij}^k, player k receives a unit loss. This differs from games with deterministic payoffs as there is uncertainty in the result based upon the strategy pair. In the model, the game payoffs are the expected difference in gain and loss, $g_{ij}^k = 2d_{ij}^k - 1$, which scales to the interval $[-1, +1]$. The bi-matrix $\mathbf{D}$ is a nonzero-sum game such that both players may receive a unit gain (or unit loss), that is, d_{ij}^1 does not necessarily equal 1-d_{ij}^2.

The decisions are made using randomization and, as such, both players are uncertain as to the pure strategy that will be employed by the other player. Let $\mathbf{p} = (p_1, p_2)$ be the state vector where p_k is the probability that player k will select strategy 1 and $\bar{p}_k$ is the probability of strategy 2. Each player employs an automaton to update the probabilities for the next stage where a unit gain is a reward and a unit loss is a penalty.

The following closely parallels the derivation in [1] except that we include nonzero-sum games, delayed information, and a modified notation.

Let $\delta z(t) = z(t+1) - z(t)$. The expected change in the probability vector can be deduced from (1). For example, with probability p_1, player 1 will select strategy 1. If the player receives a reward, then p_1 will increment by $\theta\beta\bar{p}_1$. Following this reasoning for all possibilities:

$$E[\delta\mathbf{p}(t)|\mathbf{p}(t) = \mathbf{p}] = \theta\mathbf{W}(\mathbf{p}), \quad (2)$$

where

$$W_k(\mathbf{p}) = \beta p_k \bar{p}_k [C_1^k(\mathbf{p}) \text{-} C_2^k(\mathbf{p})] + \alpha[\bar{p}_k^2 \bar{C}_2^k(\mathbf{p}) \text{-} p_k^2 \bar{C}_1^k(\mathbf{p})]$$

and $C_i^k(\mathbf{p})$ is the probability that player k receives a reward for strategy i. This is determined as follows. Let $\mathbf{p}_k = (p_k, \bar{p}_k)$ be the probability vector for player k. The expected game payoff, or value of the game, for player k is

$$\eta_k(\mathbf{p}) = \mathbf{p}_1 \mathbf{D}^k \mathbf{p}_2^T, \quad (3)$$

where $\mathbf{p}_2^T$ is the transpose of $\mathbf{p}_2$. Now, $C_i^k(\mathbf{p}) = \eta_k(\mathbf{q})$ where $\mathbf{q} = \mathbf{p}$ but with the kth element replaced by 2-i. For example, if player 1 selects strategy 1, then the expected payoff is $p_2 d_{11}^1 + \bar{p}_2 d_{12}^1$.

We recast the difference equation as a differential equation as this closely captures the behavior for the typical parameter settings, i.e. small θ. Therefore,

$$\frac{d\mathbf{p}}{dt} = \theta\mathbf{W}(\mathbf{p}). \quad (4)$$

The equilibrium solution is $\mathbf{p}^*$ where $\mathbf{W}(\mathbf{p}^*) = 0$. Note that the values of the learning parameters affect the equilibrium solution, that is, $\mathbf{p}^* = f(\alpha, \beta, \mathbf{D})$.

Step 3: Delayed Information

Since players represent agents in a physically distributed environment, the information available to a player is delayed. The state vector $\mathbf{p}$ describes the probabilities of decisions and, in our model, is subject to aged information. That is, the players must make the best decisions possible given an aged view of the likelihood of the other agent's decisions.

Let τ be the average delay in information, representing the overall effect of latency within the distributed system. For example, latency is increased by periodic broadcasts of information [4] or by the inherent delays within network hardware and software. The latency is a fundamental cause of uncertainty.

In the original model [1], each player is unaware of the other player's mixed (or even pure strategy) and is only informed of the outcome of the stage (gain or loss). The outcome is determined by both chance and the selected strategies. This implies that an umpire (or some system module) has access to both pure strategies and performs a randomizing event to determine the gain or loss for each player.

In our model, each player interacts with a local, rather than global, module to determine the outcome. The local module knows with certainty the pure strategy employed by the local player but not that of the "distant" player. Instead, the module knows the mixed strategy at τ stages earlier and employs this information to determine a new matrix of probabilities.

Consider a probability $p_k(t)$. We define an aged view of this probability as $p_k^\tau = p_k(t\text{-}\tau)$ with $p_k(t) = p_k(0)$ for $t < 0$. Player k knows with certainty the probability of its own strategy p_k and has an aged view of the other probability. Let $\mathbf{p}^k$ be player k's view of the state vector, that is, $\mathbf{p}^1 = (p_1, p_2^\tau)$ and $\mathbf{p}^2 = (p_1^\tau, p_2)$. For example, the local module for agent 1 determines the outcome based on the agent's pure strategy i and chance, but where chance is now determined by $p_2^\tau d_{i1} + \bar{p}_2^\tau d_{i2}$.

Now (4) may be applied using $W_k(\mathbf{p}^k)$ instead of the instantaneous vector $\mathbf{p}$. Formally, (4) is a nonlinear delay differential equation [8].

III. Characteristic Behaviors

The following games are considered as examples of payoffs that might be found in an application:

$$\mathbf{D}_1 = \begin{bmatrix} .6,.4 & .8,.2 \\ .35,.65 & .9,.1 \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} .75,1 & .5,.25 \\ 1,.5 & .25,.75 \end{bmatrix},$$

$$\mathbf{D}_3 = \begin{bmatrix} .4,.4 & 1,1 \\ 1,1 & 0,0 \end{bmatrix},$$

where $\mathbf{D}_1$ is a zero-sum game with a mixed strategy equilibrium [1], $\mathbf{D}_2$ is a nonzero-sum game with a mixed strategy equilibrium, and $\mathbf{D}_3$ is a nonzero-sum game with two pure strategy equilibria.

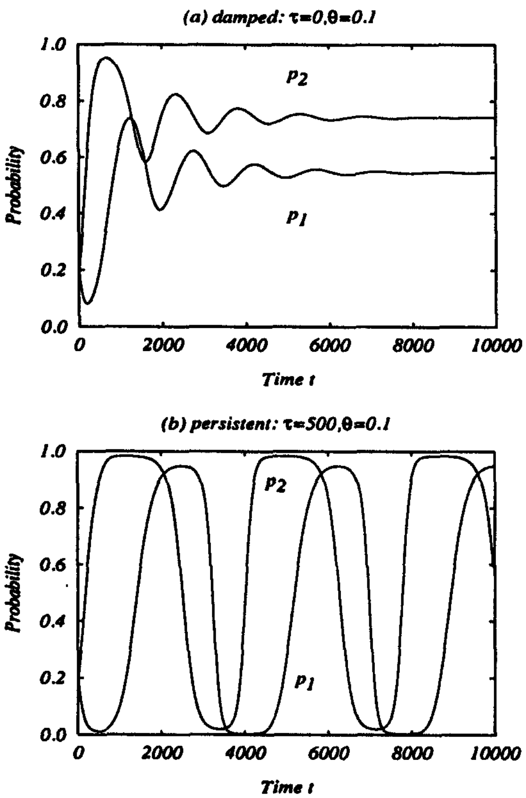


Fig. 1: Theoretical Oscillations at Low and High Delay

Two examples of the behavior of the players in $\mathbf{D}_1$, based on the above dynamical equation, are shown in Fig. 1 for a total number of stages $n = 10000$ (α=0.005, β=0.2, θ=0.1). Fig. 1(a) shows that, with no delay, damped oscillations reach an equilibrium that is very close to the optimum (as reported in [1]). However, persistent and high amplitude oscillations are predicted with high delay, Fig. 1(b).

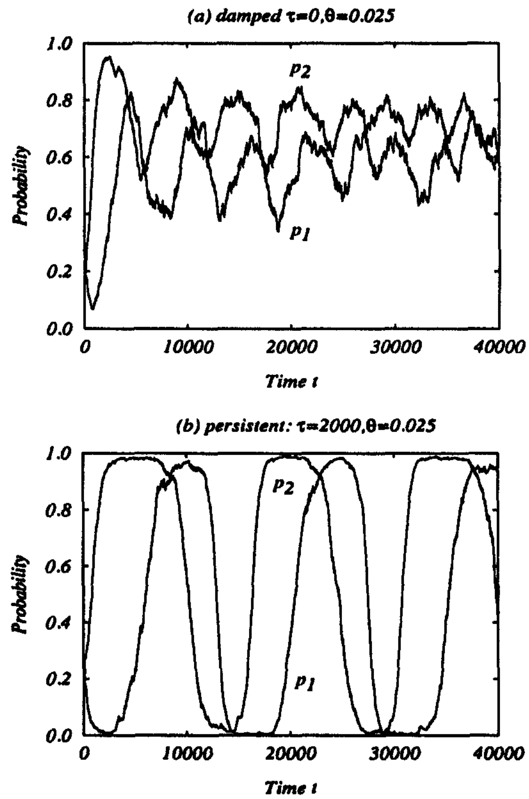


Fig. 2: Simulation at Low and High Delay

Fig. 2 shows the results of single simulation runs for the same experiment. Note that the relative delay is the same in both Fig. 2(b) and Fig. 1(b), i.e. $\theta\tau$=50, though the individual parameters differ by a factor of 4. Small step sizes are necessary to yield accurate simulation runs.

In Fig. 2(a), the oscillations do not show damped behavior but, instead, exhibit rough oscillations followed by irregular patterns. The high delay case, Fig. 2(b), agrees closely with its counterpart, Fig. 1(b). Although the simulation results show an approximate agreement with the results from the dynamical equation, multiple runs cannot be combined as small phase shifts eventually obliterate any oscillatory behavior even though each individual run shows persistent oscillations [10]. An alternative is to correlate the data within each run and average the results [10].

Fig. 3 shows the predicted behavior near the bound-

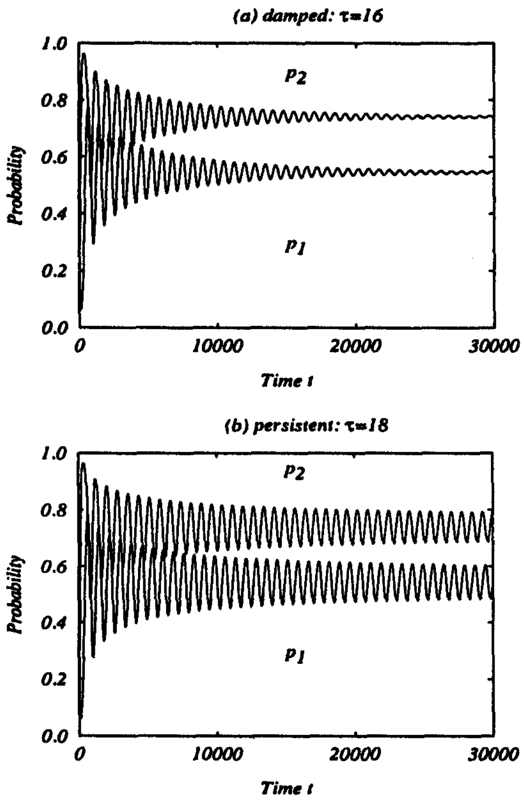


Fig. 3: Oscillations Near Stability Boundary

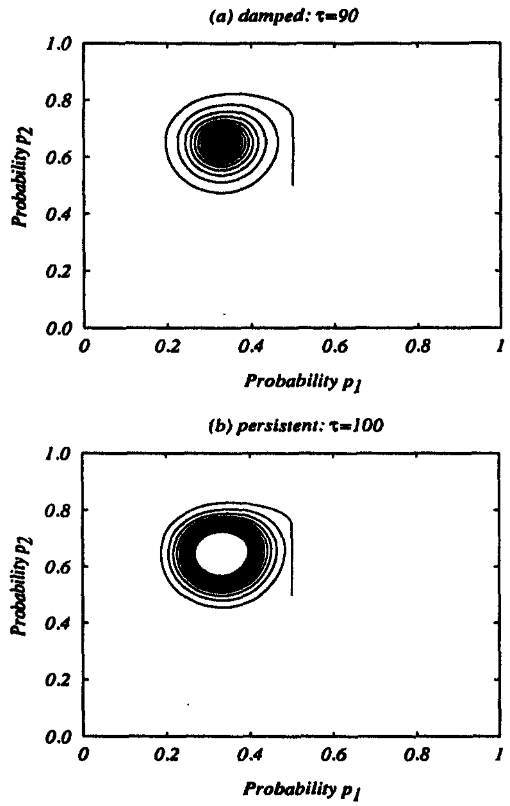


Fig. 4: Phase-Plane Portrait Near Stability Boundary

ary between damped and persistent oscillations (α=0.01, β=0.4, θ=0.1). In Fig. 3(a), τ=16 and the mixed strategies exhibit damped oscillations which slowly settle into an equilibrium. However, in Fig. 3(b), a small increase in τ (18) initiates persistent oscillations. We can say that the delay τ_2 required to initiate persistent oscillations is $16 < \tau_2 \leq 18$.

Fig. 4 shows phase-plane portraits, rather than time-dependent behavior, for $\mathbf{D}_2$. In the damped case, Fig. 4(a), the oscillations reach an equilibrium such that the center of the spiral vanishes while persistent oscillations, Fig. 4(b), circulate about the attractor equilibrium. We observe that $90 < \tau_2 \leq 100$.

Fig. 5(b) shows, theoretically, that the players oscillate synchronously between the two pure strategy equilibria in $\mathbf{D}_3$ for the given initialization. However, Fig. 5(a) shows that small phase shifts in the simulation eventually allow the players to reach an equilibrium in pure strategies.

The average mixed strategy (n=30000) is shown in Fig. 6 for two cases of β in $\mathbf{D}_1$. For τ=0 (solid lines), the average strategies approach the optima for low α, that is, the players learn the optimal strategies for sufficiently small parameter values (as in [1]). At a fixed α, a smaller value of β degrades the performance - it is best to have a large distance between the parameter values. Two examples of delay (dotted lines) are also shown for $\beta = 0.2$. As the delay increases, the performance decreases, however, there is a sufficient value of α for which the behavior approximates the instantaneous case. At the point where the delay and instantaneous cases agree, the persistent oscillations give way to damped oscillations, hence a stable, but suboptimal, equilibrium.

As the delay goes to zero, the only unstable parameter setting is α=0, which is the linear reward-inaction algorithm (L_{R-I}). This has been shown to oscillate due to non-negative real parts in the eigenvalues of the Hessian of the non-delayed version of (4) [1].

The characteristics of the oscillations, based on various parameter settings, are shown in Fig. 7 for $\mathbf{D}_1$. The normalized frequency of oscillation $\nu = \tau/T$, where T is the period of oscillation or time between successive maxima. The normalized frequency increases with delay τ and learning parameters α and β. Note that one particular instance (α=0.01, β=0.2) does not exhibit a

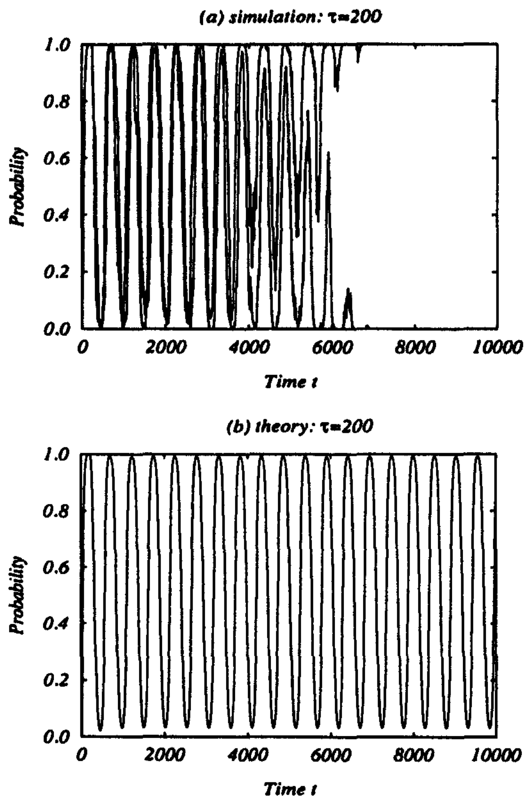


Fig. 5: Randomization in Simulation Leads to Equilibrium

sufficient number of maxima to determine the characteristics at low delay.

Fig. 7(b) shows the amplitude or extent of the oscillations in player 1's mixed strategy. The normalized amplitude s is the root mean square:

$$s = \sqrt{\Sigma_{t=1}^{n}(p_1(t) - P_{opt})^2/n},$$

which increases with delay τ and reward parameter β, but decreases with penalty parameter α. (Note that we measure the oscillations about the consistent value of the optimum rather than the average value for the particular parameter.) Fig. 7 does not indicate the effect of the step size parameter on the oscillations but ν and s increase with θ.

IV. Stability Boundary

The goal is to predict, using linear stability analysis [10], the delay τ_2 sufficient to initiate persistent oscillations. Let $k \in \{1, 2\}$ and

$$X_k = \theta\frac{\partial W_k(\mathbf{p}^*)}{\partial p_k}, \quad Y_k = \theta\frac{\partial W_k(\mathbf{p}^*)}{\partial p_{3-k}^{\tau}}.$$

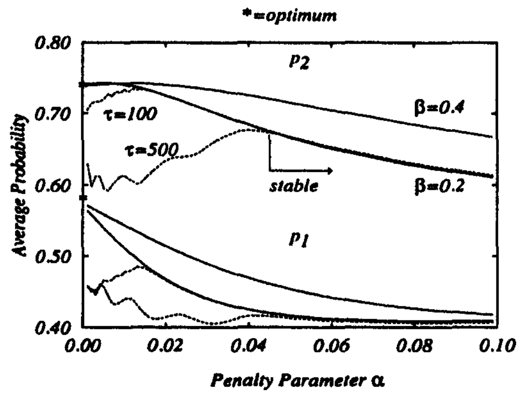


Fig. 6: Effects of Learning and Delay on Average Strategy

Linearizing in the neighborhood of the equilibrium:

$$\frac{d\delta p_k}{dt} = X_k\delta p_k + Y_k\delta p_{3-k}^{\tau}. \tag{5}$$

Assuming a solution of the form $\delta p_k(t) = A_k e^{\lambda t}$,

$$(\lambda - X_1)(\lambda - X_2) = Y_1Y_2e^{-2\lambda\tau}. \tag{6}$$

Let $\lambda = r + iw$. The stable solutions are determined by the parameter settings that yield only negative real parts. That is, marginal stability occurs at r=0. Solving (6) for the real and imaginary parts:

$$\cos(2w\tau) = (X - w^2)/Y, \tag{7}$$
$$\sin(2w\tau) = (X_1 + X_2)w/Y, \tag{8}$$

respectively, where $X = X_1X_2$ and $Y = Y_1Y_2$. Note that (7) and (8) are an implicit solution for τ_2 (an analytic solution: sum the squares of (7) and (8) to solve for w and divide (8) by (7) to solve for τ in terms of w).

Fig. 8 shows the instability delay, $\tau_2 = f(\alpha, \beta, \theta = 0.1, \mathbf{D}_1)$, that results from two types of calculations. First, individual data points represent the onset of oscillations as determined by (4), the strategies are examined over long runs at incremental delays to determine which delay is sufficient. Second, the dotted lines are the solution to the predicted behavior (7) and (8) and approximate the numerical results to within 5%. In general, the delay required to initiate persistent oscillations increases with α and decreases with β. This is exactly *opposite* the conditions that insure an equilibrium that is close to the value of the game (Fig. 6 and [1]). That is, parameters that are likely to lead to the optimal equilibrium in the non-delayed case are more likely to initiate instabilities in the delayed case.

The effect of θ is not shown but the instability delay is inversely proportional to θ, since the players do not make

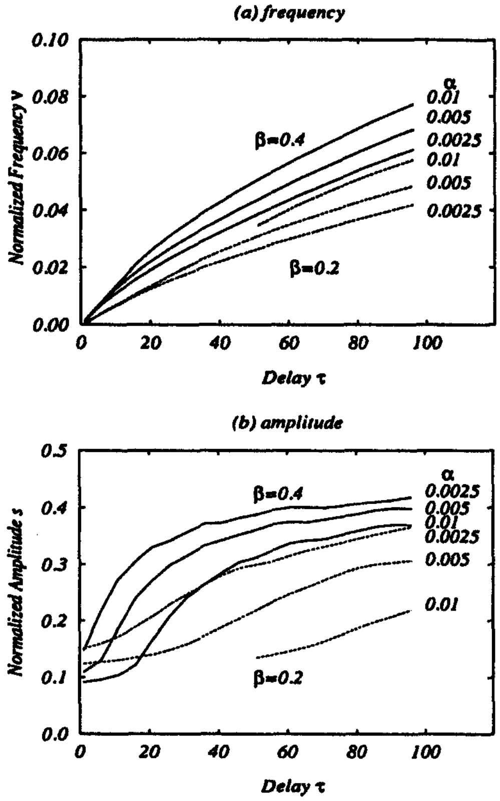


Fig. 7: Effects of Learning Parameters and Delay

large steps in their decision-making process. However, a small θ also means that the players take longer to reach equilibrium.

Fig. 8 also suggests that the data for the $\beta = 0.1$ and $\beta = 0.2$ cases are approaching asymptotes. This is true of the other cases, though not shown in the figure. That is, there is a sufficient α for a given β such that the instability delay τ_2 goes to infinity. Beyond this value of α, the system is observed to be stable albeit at the loss of an equilibrium close to the value of the game.

V. Conclusions

Learning automata have been examined in many theoretical and applied studies, typically demonstrating good equilibrium properties. In mixed strategy games with delayed information, however, the automata may exhibit unstable behavior. The selection of a small α, to insure an equilibrium close to the value of the game, makes the system less tolerant of delays. An alternative is to select a large α, thus insuring a stable but less optimal, equilibrium. Linear stability analysis determines the boundary between stable and unstable systems.

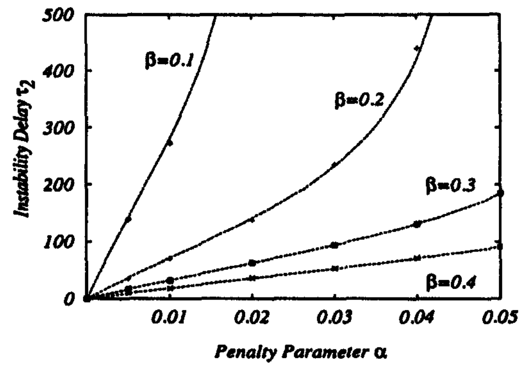


Fig. 8: Stability Boundary

References

[1] S. Lakshmivarahan and K. Narendra, "Learning algorithms for two-person zero-sum stochastic games with incomplete information:A unified approach," *SIAM J. Control and Optimization*, vol. 20, no. 4, pp. 541–552, July 1982.

[2] K. Narendra and M. Thathachar, *Learning Automata: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] O. Nedzelnitsky and K. Narendra, "Nonstationary models of learning automata routing in data communication networks," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, no. 6, pp. 1004–1015, Nov./Dec. 1987.

[4] R. Mirchandaney and J. Stankovic, "Using stochastic learning automata for job scheduling in distributed processing systems," *J. Parallel and Distr. Comput.*, vol. 3, pp. 527–552, 1986.

[5] A. Glockner and J. Pasquale, "Coadaptive behavior in a simple distributed job scheduling system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 902–907, May/June 1993.

[6] E. Billard and J. Pasquale, "Effects of delayed communication in dynamic group formation," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1265–1275, Sept./Oct. 1993.

[7] E. Billard and J. Pasquale, "Dynamic scope of control in decentralized job scheduling," in *Proc. IEEE Intl. Symp. Autonomous Decentralized Syst.*, pp. 183–189, Mar. 1993.

[8] J. Wiener and J. Hale, eds., *Ordinary and Delay Differential Equations*. New York: John Wiley & Sons, 1992.

[9] T. Hogg and B. A. Huberman, "Controlling chaos in distributed systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 1325–1332, Nov./Dec. 1991.

[10] J. O. Kephart, T. Hogg, and B. A. Huberman, "Dynamics of computational ecosystems," *Phys. Rev. A*, vol. 40, pp. 404–421, July 1989.

Learning in Multi-Level Stochastic Games with Delayed Information

Edward A. Billard
Faculty of Computer Science and Engineering
University of Aizu
Aizu-Wakamatsu City, 965-80 Japan
E-mail: billard@u-aizu.ac.jp

Abstract

Distributed decision makers are modeled as players in a game with two levels. High level decisions concern the game environment and determine the willingness of the players to form a coalition (or group). Low level decisions involve the actions to be implemented within the chosen environment. Coalition and action strategies are determined by probability distributions which are updated using learning automata schemes. The payoffs are also probabilistic and there is uncertainty in the state vector since information is delayed. The goal is to reach equilibrium in both levels of decision making; the results show the conditions for instability, based on the age of information.

1 Introduction

Agents in a distributed system make decisions to optimize a performance metric or achieve a more abstract set of goals. These agents must typically consider working with other agents to cooperatively achieve the desired result. However, there is a high degree of uncertainty in these activities. First, the agent may not know the true state of the system as a result of delayed information. The delays may be due to inherent latencies in a network or the intermittent (or periodic) exchange of information. The agents make the best possible decisions with the information available [Gmytrasiewicz *et al.*, 1991c]. Second, even with instantaneous information, there is uncertainty in the strategies employed by the other agents given the state vector. For example, an agent may not be certain that another agent is willing to cooperate or to what extent. Third, even with knowledge of the other strategies, there is uncertainty in the payoffs that result from the combined actions.

We present a model to capture the nature of these various uncertainties with distributed decision makers as players in a game with two levels. The high level concerns the game environment and determines the willingness of the players to form a coalition (or group). The low level involves the actions to be implemented within the chosen environment.

Both of these strategies are modeled using probability distributions with updates according to learning automata schemes [Narendra and Thathachar, 1989]. This implies that learning is taking place on two levels and a constraint is that a player must make both decisions simultaneously, without knowledge of the other players' decisions at either level. In particular, a player knows whether it is willing to form a group but does not know the intentions of the other players. This implies that a player may select an action under the assumption of cooperative behavior but this action, in the context of non-cooperative behavior, may result in suboptimal performance.

The payoffs in the games are stochastic, that is, there is a probability of gain or loss based upon the action set. Uncertainty in information is captured by the assumption that an average age of information exists in the system. The goal of the model is to capture decision making under uncertainty in various domains and to summarize uncertainty as probability distributions. The adaptive learning schemes easily model the uncertainty, permit expected value computations to determine beliefs, and have analytic solutions to complex dynamical behaviors. These schemes may also be considered as approximations to more complex reasoning schemes.

In most distributed systems an important goal is to achieve a stable solution. We develop a dynamical equation to predict the behavior based on the parameter settings and apply linear stability analysis to predict the onset of persistent oscillations.

The paper is organized as follows: Section 2 describes related work; Section 3 develops the model in stages, including the dynamical equation; Section 4 shows example simulations and associated predicted behavior. In Section 5, we make an assumption that leads to a reasonably accurate prediction of the delay required to initiate persistent instabilities in the system. Our conclusions are presented in Section 6.

2 Related Work

Our interests in distributed decision making are closely related to the evolution of cooperation [Axelrod and Hamilton, 1981] and computational ecosystems. The original description of computational ecologies [Huberman and Hogg, 1988] shows the dynamical equation based on simple gain functions with imperfect and delayed information. A large system of agents select resources based on aged information of other agents' resource preferences. The resultant behavior can be categorized as stable, oscillatory (both damped and persistent), or chaotic (with possible bifurcations). The agreement between the dynamical equation and simulation is demonstrated in [Kephart *et al.*, 1989] and the existence of a general adaptive strategy to eliminate the instabilities is shown in [Hogg and Huberman, 1991].

In distributed computing systems, a high degree of physical decentralization leads to aged information such that agents are not able to attain common knowledge [Halpern and Moses, 1990]. The goal of agents in these systems is to make good decisions with the information available and, in particular, to make good decisions involving cooperation with other agents. Other research examines cooperation without communication [Genesereth *et al.*, 1985] and cooperation with negotiated protocols [Rosenschein and Genesereth, 1985].

Our approach is to examine learning mechanisms such as learning automata [Narendra and Thathachar, 1989] in environments with delayed information. The basic research relevant to automata playing stochastic games (and the associated dynamics) is found in [Lakshmivarahan and Narendra, 1982]. Our model extends this to delayed information and a hierarchy of games. The games in our model represent the payoffs of an underlying application such as robotics [Gmytrasiewicz *et al.*, 1991a].

Learning automata have demonstrated coadaptive behavior in a distributed queueing system [Glockner and Pasquale, 1993]. We have also examined learning automata in autonomous decentralized queueing systems [Billard and Pasquale, 1993a] and in games [Billard and Pasquale, 1993b]. We view the learning algorithms as generic in the sense that they capture incremental, or adaptive, learning.

Although increased levels of communication can reduce the age of information to the minimum latency, there is an associated cost in processing this information. For this reason, it is important to exchange only the appropriate information. This can be done based on expected utility [Gmytrasiewicz *et al.*, 1991c] with agents reaching equilibrium using recursive reasoning [Gmytrasiewicz *et al.*, 1991b].

3 The Model

The model is developed in four stages: 1) the basic algorithm for a learning automaton [Narendra and Thathachar, 1989], 2) the algorithm applied to the strategies of two players in a game, 3) the algorithm applied again to the strategies of selecting between two games, and 4) the delay in state information. The salient feature of the model is that each agent makes a decision to work in a group or alone, thus affecting the environmental payoffs, and a decision regarding the action to be taken within the chosen environment.

Step 1: One Automaton - Two Strategies

Let $p(t)$ and $\bar{p}(t)$ be the probability of selecting strategy 1 and strategy 2, respectively, at time t. The probability is incremented or decremented for the next time step by

$$\Delta p = \theta \cdot \begin{cases} +\beta\bar{p} & \text{if reward on strategy 1} \\ -\beta p & \text{if reward on strategy 2} \\ -\alpha p & \text{if penalty on strategy 1} \\ +\alpha\bar{p} & \text{if penalty on strategy 2} \end{cases} . \quad (1)$$

The extent of the incremental change in the mixed strategy is determined by the three constants: β is the reward parameter, α is the penalty parameter, and θ is the step size parameter. It is assumed that $0 < \alpha < \beta < 1$ and $0 < \theta \leq 1$. Although θ can be incorporated into α and β, it is convenient to extract this term for simulation and analysis results.

Step 2: Two Players - Two Strategies

We define two players $k, l \in \{1,2\}$ in a game $\mathbf{D} = (\mathbf{D}^1, \mathbf{D}^2)$, where $\mathbf{D}^k$ represents a stochastic payoff matrix for player k [Lakshmivarahan and Narendra, 1982; Narendra and Thathachar, 1989] and corresponds to an underlying application. Each player chooses a strategy $i, j \in \{1,2\}$, respectively, and the game is played in stages with element d_{ij}^k of $\mathbf{D}^k$ being the probability of a unit gain for player k based upon the strategy pair (i,j). With probability 1-d_{ij}^k, player k receives a unit loss. This differs from games with deterministic payoffs as there is uncertainty in the result based upon the strategy pair. In the model, the game payoffs are the expected difference in gain and loss, $g_{ij}^k = 2d_{ij}^k - 1$, which scales to the interval $[-1,+1]$. The bi-matrix $\mathbf{D}$ is a nonzero-sum game such that both players may receive a unit gain (or unit loss), that is, d_{ij}^1 does not necessarily equal 1-d_{ij}^2.

The decisions are made using randomization and, as such, both players are uncertain as to the pure strategy that will be employed by the other player. Let $\mathbf{p} = (p_1, p_2)$ be the state vector where p_k is the probability that player k will select strategy 1 and $\bar{p}_k$ is the probability of strategy 2. Each player employs an automaton to update the probabilities for the next stage where a unit gain is a reward and a unit loss is a penalty.

The following closely parallels the derivation in [Lakshmivarahan and Narendra, 1982] except that we include nonzero-sum games, delayed information, and a more general notation that permits learning in a hierarchy of games.

Let $\delta z(t) = z(t+1) - z(t)$. The expected change in the probability vector can be deduced from (1). For example, with probability p_1, player 1 will select strategy 1. If the player receives a reward, then p_1 will increment by $\theta\beta\bar{p}_1$. Following this reasoning for all possibilities:

$$E[\delta\mathbf{p}(t)|\mathbf{p}(t) = \mathbf{p}] = \theta\mathbf{W}(\mathbf{p}), \quad (2)$$

where

$$W_k(\mathbf{p}) = \beta p_k\bar{p}_k[C_1^k(\mathbf{p}) - C_2^k(\mathbf{p})] + \alpha[\bar{p}_k^2\bar{C}_2^k(\mathbf{p}) - p_k^2\bar{C}_1^k(\mathbf{p})] \quad (3)$$

and $C_i^k(\mathbf{p})$ is the probability that player k receives a reward for strategy i. This is determined as follows. Let $\mathbf{p}_k = (p_k \ \bar{p}_k)$ be the probability vector for player k. The expected game payoff, or value of the game, for player k is

$$\eta_k(\mathbf{p}) = \mathbf{p}_1\mathbf{D}^k\mathbf{p}_2^T, \quad (4)$$

where $\mathbf{p}_2^T$ is the transpose of $\mathbf{p}_2$. Now, $C_i^k(\mathbf{p}) = \eta_k(\mathbf{q})$ where $\mathbf{q} = \mathbf{p}$ but with the kth element replaced by 2-i. For example, if player 1 selects strategy 1, then the expected payoff is $p_2d_{11}^1 + \bar{p}_2d_{12}^1$.

We recast the difference equation as a differential equation as this closely captures the behavior for the typical parameter settings, i.e. small θ. Therefore,

$$\frac{d\mathbf{p}}{dt} = \theta\mathbf{W}(\mathbf{p}). \quad (5)$$

The equilibrium solution is $\mathbf{p}^*$ where $\mathbf{W}(\mathbf{p}^*) = 0$. Note that the values of the learning parameters affect the equilibrium solution, that is, $\mathbf{p}^* = f(\alpha, \beta, \mathbf{D})$.

Step 3: Four Players - Two Games

We introduce the concept of multi-level games to capture the notion of cooperation in group dynamics, see Figure 1. An agent consists of two subcomponents, or players, each of which is modeled as a learning automaton. One player within each agent makes a preference decision between two game bi-matrices **A**, the *non-default* game matrix, and **B**, the *default* game matrix. Game **B** represents the underlying environment when the agents choose not to form a group. Typically, the payoffs will be lower but easier to achieve (in the sense of an equilibrium). Game **A** represents the environment when both agents agree to cooperate in a group with the expectation that better payoffs are available to both agents. However, to achieve these payoffs, the agents must successfully coordinate their actions within the game, perhaps a more difficult task in this game than in **B**. This second activity, i.e. selecting an action strategy within the chosen game environment, is carried out by an additional player within each agent. If an agent is willing to play game **A**, there is uncertainty whether the other agent will agree and, hence, the player subcomponent may make poor action decisions. For example, player 1 may select action strategy 1 since it has a high expectation of success in game **A**, the agent's preferred matrix. If agent 2 forces the default game environment, strategy 1 may yield a very poor result. It is the uncertainty in coalition formation and the simultaneity of decision making that makes action decision making difficult.

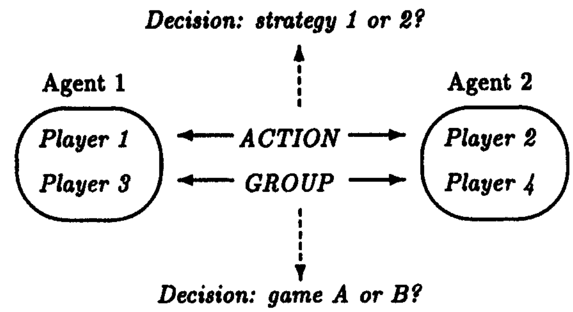


Figure 1: Multi-Level Decision Making by Agents' Components

We define the high level decisions (i.e. *which* game matrix) as group strategies and the low level decisions (i.e. which strategy *within* a game) as action strategies. The formal definition of the model is as follows.

The action strategies are determined as before (using p_1 and p_2). The group strategies are also made using randomization with p_3 the probability that player 3 (a subcomponent of agent 1) will prefer **A** over **B** (likewise, p_4 is the probability for player 4, a subcomponent of agent 2). The state vector is now $\mathbf{p} = (p_1, p_2, p_3, p_4)$. At the high level, each player uses an automaton to decide the game preference. At the low level, each player uses a different automaton to select a strategy. The action pair is determined at the same time as the group decision. The resultant action pair (i, j) is played in game **A** if, and only if, *both* agents prefer this game matrix. That is, the agents agree to form a coalition with probability $c = p_3p_4$, the clustering parameter. Otherwise, the stochastic payoffs are determined by **B** with the agents operating in a non-coalition mode. The problem of apportioning credit to the different levels is avoided by assuming that both levels receive the same payoff, that is, both receive either a unit gain or unit loss.

An average game is induced based on the high level strategies:

$$\mathbf{D}^k = c \cdot \mathbf{A}^k + (1-c) \cdot \mathbf{B}^k. \quad (6)$$

The dynamical equation is still (5) but where $k \in \{1, 2, 3, 4\}$ and $\mathbf{D}^k = \mathbf{D}^{k-2}$ for $k \in \{3, 4\}$. Note that this equation enforces a strong interaction among the state variables. The low level strategies are dependent on the high level strategies for the expectation of the

A_1		B_1		A_2		B_2		A_3		B_2	
1,1	0,0	0,0	0,0	.6,.4	.2,.8	.4,.6	.8,.2	.75,1	.5,.25	.4,.6	.8,.2
0,0	0,0	0,0	1,1	.35,.65	.9,.1	.65,.35	.1,.9	1,.5	.25,.75	.65,.35	.1,.9
game 1				*game 2*				*game 3*			

Figure 2: Example Games

average game. They are also dependent on each other via the stochastic payoffs based on action pairs. The high level strategies are dependent on the low level strategies since the reward (or penalty) is derived in the same way. The potential exists for different learning rates at different levels but, in this study, both rates are identically θ.

Step 4: Delayed Information

Since agents are physically distributed, the information available to an agent is delayed. The state vector $\mathbf{p}$ describes the probabilities of decisions at both the high and low level and, in our model, is subject to aged information. That is, the agents must make the best decisions possible given an aged view of the likelihood of the other agent's decisions.

Let τ be the average delay in information, representing the overall effect of latency within the distributed system. For example, latency is increased by periodic broadcasts of information or by the inherent delays within network hardware and software. The latency is a fundamental cause of uncertainty.

Consider a probability $p_k(t)$. We define an aged view of this probability as $p_k^\tau = p_k(t\text{-}\tau)$ where $p_k(t) = p_k(0)$ for $t < 0$. Agent k knows with certainty the probability of its low and high strategies, p_k and p_{k+2}, respectively, and has an aged view of the other two probabilities. From the subcomponents point of view, let $\mathbf{p}^k$ be player k's view of the state vector, that is, $\mathbf{p}^1 = \mathbf{p}^3 = (p_1, p_2^\tau, p_3, p_4^\tau)$ and $\mathbf{p}^2 = \mathbf{p}^4 = (p_1^\tau, p_2, p_3^\tau, p_4)$.

In terms of the rules of the game, the preceding implies that a local module, or score keeper, provides a unit gain or loss based on the decisions of the local agent and the aged probabilities of the distant agent. For example, the local module for agent 1 determines the outcome based on the agent's pure strategy i and chance, but where chance is now determined by $p_2^\tau d_{i1} + \bar{p}_2^\tau d_{i2}$ (and the average game element d_{ij} is also based on aged information).

Now, (5) may be applied using $W_k(\mathbf{p}^k)$ instead of the instantaneous vector $\mathbf{p}$. For example, the rate of change in $p_1(t)$ is a function of $p_1(t)$, $p_3(t)$, $p_2(t\text{-}\tau)$, and $p_4(t\text{-}\tau)$. Formally, (5) is a nonlinear delay differential equation [Wiener and Hale, 1992].

4 Experiments

Three games, see Figure 2, are considered with respect to learning behavior and the stability of the probabilistic strategies. The games are chosen to facilitate the illustration of key points and do not necessarily represent an underlying application. In game 1, the high level choice is between two game matrices, both with pure strategy equilibria of identical payoffs to both players. However, an opposite set of actions is required to achieve equilibrium. In game 2, the matrices are complements of each other and both are zero-sum game matrices with mixed strategy equilibria (the single game is from [Lakshmivarahan and Narendra, 1982]). In game 3, one choice is a nonzero-sum game matrix with mixed strategy equilibrium and the other is the same default game matrix of game 2.

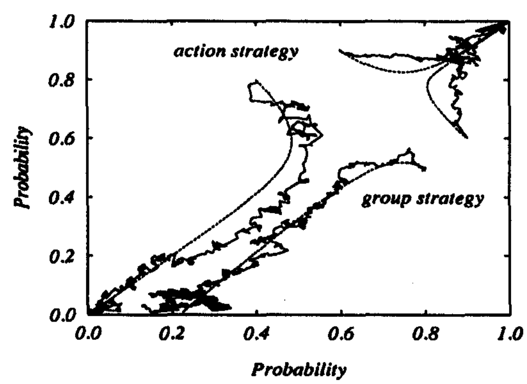


Figure 3: Phase-Plane Portrait of Game 1 with Two Pure Strategy Equilbria ($\tau = 0$)

Figure 3 shows the action and group strategies for game 1 in two experiments with different initialization (the delay in the system is zero.) The action strategies are plotted as p_2 versus p_1 and the group strategies as p_4 versus p_3. The initialization determines which of the two pure equilibria is "closest". The single runs roughly approximate the predicted behavior based on a numerical solution to (5), that is, the players are able to reach an equilibrium in both levels. Note that the group strategy for the non-coalition equilibrium does not terminate at the origin. Instead, both strategies decrease at the same linear rate and whichever strategy reaches zero first (based on initialization) prevents

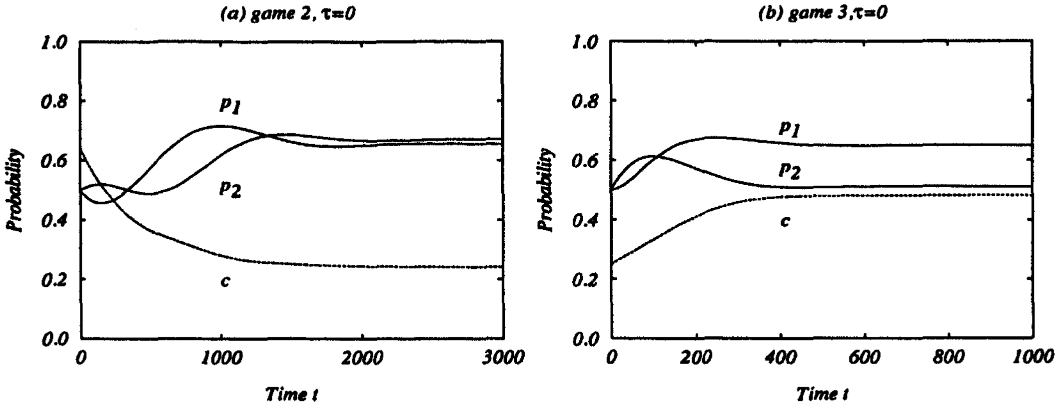


Figure 4: Equilibria in Action and Group Mixed Strategies

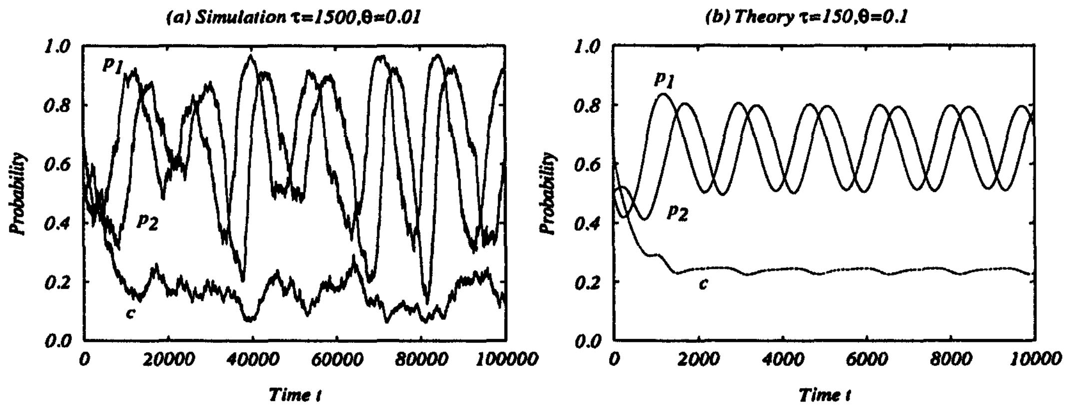


Figure 5: Oscillations in Action Strategies for Game 2

coalition formation (i.e. $c=p_3p_4=0$). This linear behavior is due to the contrived nature of the game payoffs.

Figure 3 also shows that a large region of initialization is expected to result in the non-coalition equilibrium. For example, initialization $p_3=p_4=0.7$ is in the upper-right corner but is actually slightly biased to the non-coalition equilibrium ($c=p_3p_4=0.49$).

Figure 4 shows that players (theoretically) are able to reach mixed strategy equilibria (in both the group and action levels) for games 2 and 3 without delays. (Unless otherwise stated, $\alpha=0.02, \beta=0.4, \theta=0.01$ for game 2 and $\alpha=0.01, \beta=0.05, \theta=0.1$ for game 3). A distinction between the two games is that the likelihood of group formation, as defined by the clustering parameter c, decreases in game 2 and increases in game 3. We now consider the effects of delays in the information exchanged in these two games.

Figure 5 shows both a single simulation run and the prediction of (5) for game 2. In experiments with instabilities, it is not typically possible to combine multiple runs [Kephart *et al.*, 1989]. Each particular run may grossly approximate theory, for example, by displaying persistent oscillations of appropriate amplitude and frequency. However, there are small phase shifts in the oscillations among multiple runs that lead to eventual obliteration of the oscillations after a long time. Instead, correlation must be attempted within each run and then averaged over multiple runs. We do not attempt to prove the correlation here but concentrate on the predictions of the theory. Independent of the accuracy of the dynamical equation with respect to the learning automata experiment, we consider the equation to be a paradigm for incremental learning.

We note that the accuracy is affected by the learning rate θ: the smaller the learning rate, the better the accuracy. Large step sizes allow the strategies to overshoot the maxima and minima predicted. In Figure 5(a), the amplitudes in the simulation are larger than predicted but would be reduced if a smaller parameter value was chosen. In both cases, the relative delay is the same, i.e. $\theta\tau = 15$, although the individual parameters in the two cases differ by an order of magnitude. For this reason, we may examine the theory with any value of θ, though we know that a small θ must be chosen to get an accurate simulation. Note that persistent oscillations are predicted (for the action

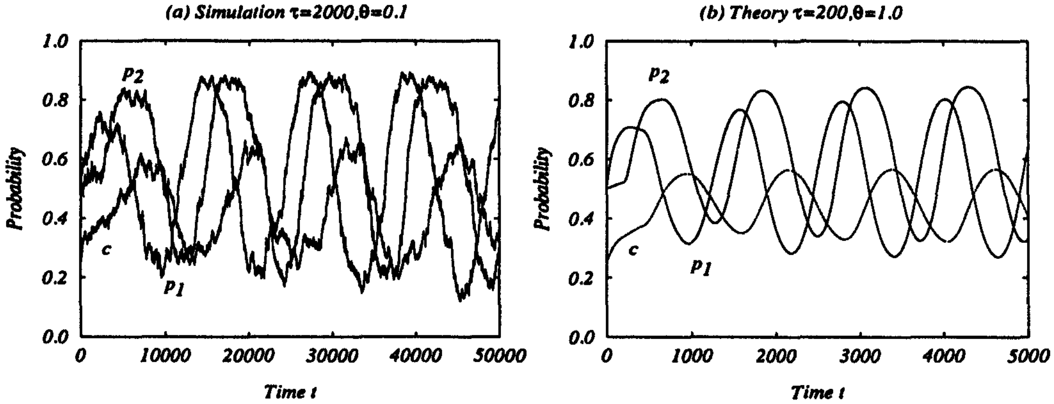


Figure 6: Oscillations in Action and Group Strategies for Game 3

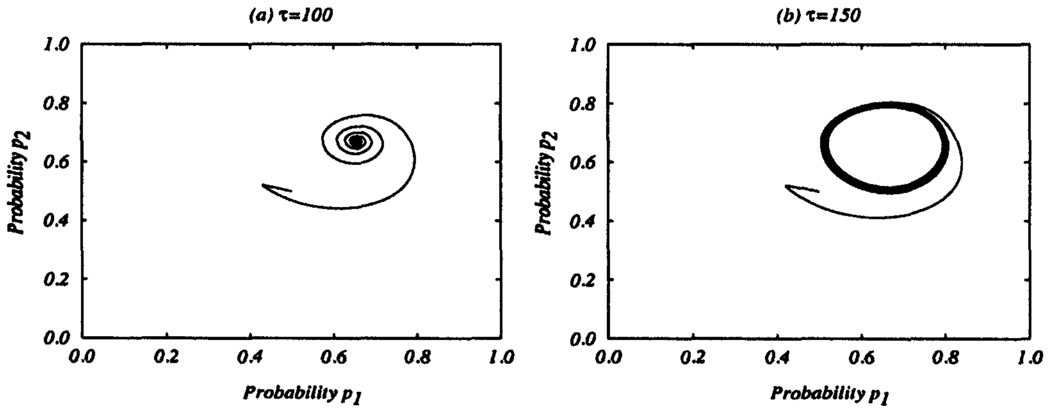


Figure 7: Phase-Plane Portraits of Action Strategies for Game 2 Near Stability Boundary

strategies) and we can say that the delay to initiate such oscillations, τ_2, must be less than or equal to 150 (for θ=0.1). At lower values of delay, the theoretical strategies exhibit damped oscillations, however, simulations do not typically show the theoretical damping but rather noise in the strategies.

Figure 5 shows that the players reach a rough equilibrium in the group strategies for game 2 but Figure 6, for game 3, shows that the group strategies oscillate persistently. In this case, we can say that $\tau_2 <= 200$ for $\theta = 1.0$. (Other experiments with this game suggest that both the action and group strategies initiate oscillatory behavior at the same delay.) There is a rough approximation between theory and simulation, again with slightly higher amplitudes in simulation due to the step size parameter.

Figure 7 shows the predicted behavior of the action strategies for game 2 for two delays near the stability boundary between damped and persistent oscillations. The damped oscillations reach an equilibrium such that the center of the spiral vanishes and the same equilibrium serves as an attractor in the persistent oscillation case (i.e. limit cycle). Note that the circular nature of the phase-plane portrait in Figure 7(b) is an alternative display of the persistent oscillations, shown over time, in Figure 6(b). From Figure 7(a) and (b), we can conclude that $100 < \tau_2 \leq 150$.

Figure 8 shows the onset of a chaotic attractor, with corresponding shifting behaviors, at very high delay. As noted in Section 3, there is a complex interaction between the two levels of learning: action strategies affect group strategies and vice versa. The high delay in the experiment induces the strategies to revisit a variety of potential equilibria, but with small shifts in the trajectory. Figure 8(a) shows the specific behavior of the action strategies and Figure 8(b) shows the behavior of the group strategies. Together, these two figures demonstrate, in four-dimensional space, the complex dynamics of learning at two levels under the circumstance of delayed information.

5 Analysis

In this section, an approximation is used to determine the amount of delay τ_2 required to initiate persistent oscillations. The technique involves linearizing in the

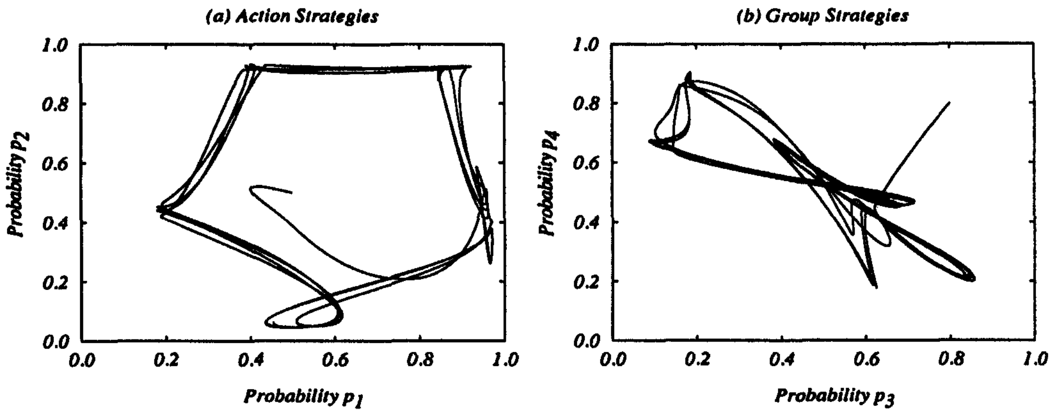


Figure 8: Chaotic Regime for Game 2 at High Delay ($\tau = 1000$)

neighborhood of the equilibrium $\mathbf{p}^*$ (a common approach [Kephart *et al.*, 1989; Farmer, 1982]) and the assumption that p_3 and p_4 are constant and equal to the equilibrium values in $\mathbf{p}^*$, that is, $c = c^*$. This implies that we ignore the partial derivatives with respect to these variables. The resultant equations are

$$\frac{d\delta p_1}{dt} = X_1 \delta p_1 + Y_1 \delta p_2^\tau \qquad (7)$$

$$\frac{d\delta p_2}{dt} = X_2 \delta p_2 + Y_2 \delta p_1^\tau \qquad (8)$$

where

$$X_1 = \theta \frac{\partial W_1(\mathbf{p}^*)}{\partial p_1},\ X_2 = \theta \frac{\partial W_2(\mathbf{p}^*)}{\partial p_2},$$

$$Y_1 = \theta \frac{\partial W_1(\mathbf{p}^*)}{\partial p_2^\tau},\ Y_2 = \theta \frac{\partial W_2(\mathbf{p}^*)}{\partial p_1^\tau}.$$

The partial derivatives are straight-forward (see [Lakshmivarahan and Narendra, 1982] for a non-delay zero-sum version).

Assuming an exponential solution of the form $\delta p_1(t) = A_1 e^{\lambda t}$, $\delta p_1^\tau(t) = A_1 e^{\lambda(t-\tau)}$, etc. yields

$$(\lambda - X_1)(\lambda - X_2) = Y_1 Y_2 e^{-2\lambda\tau}. \qquad (9)$$

Let $\lambda = r + iw$. There are an infinite number of discrete solutions and those parameter settings that yield only negative real parts are stable (with perhaps damped, but not persistent, oscillations). That is, marginal stability occurs at $r = 0$. The stability boundary can be determined by substituting $\lambda = iw$ in (9), applying Euler's formula, and solving for the real and imaginary parts:

$$\cos(2w\tau) = (X - w^2)/Y, \qquad (10)$$

$$\sin(2w\tau) = (X_1 + X_2)w/Y, \qquad (11)$$

respectively, where $X = X_1 X_2$ and $Y = Y_1 Y_2$.

Dividing (11) by (10),

$$\tan(2w\tau) = \chi = \frac{(X_1 + X_2)w}{X - w^2} \qquad (12)$$

and the instability delay, sufficient to initiate persistent oscillations, is:

$$\tau = \tau_2 = \tan^{-1}(\chi)/2w, \qquad (13)$$

where the inverse tangent takes its value in the interval $[0, \pi/2]$.

Adding the squares of (10) and (11),

$$u^2 + Bu + C = 0, \qquad (14)$$

where

$$u = w^2,\ B = X_1^2 + X_2^2,\ C = X^2 + Y^2,$$

hence $w = \pm\sqrt{u}$. The single solution to the quadratic equation is

$$u = \frac{-B + \sqrt{B^2 - 4C}}{2}, \qquad (15)$$

as the other solution fails to insure a real (the only type of solution) for w (note that $B > 0$).

We are now in a position to predict the stability boundary between damped and persistent oscillations, the results are shown in Table 1 with the cases from Figures 5 and 6 included. The predicted values τ_p are based on (13). The observed values τ_o are not from simulation but from long runs of (5) at incremental delay to determine which delay is sufficient to initiate persistent oscillations to within a high degree of accuracy. There is close agreement between the values and we can draw three simple conclusions: τ_2 increases with increasing α, decreasing β, and decreasing θ. The first two involve the relative strengths of the penalty and reward parameters. The adjustment of parameters to avoid instabilities under delayed information is exactly ***opposite*** the adjustments required to insure equilibria close to the optimal value of the game in a non-delayed environment. The last case is obvious from the fact that $\theta\tau$ is a measure of the relative delay; in fact, the table shows that the delay τ is doubled as the step size θ is halved. Finally, the data suggests that ignoring the partial derivatives with respect to p_3 and p_4 did not hinder the analytic prediction (even though these probabilities oscillated in game 3).

Table 1: Instability Delay: Predicted (τ_p) versus Observed (τ_o)

game	α	β	θ	c^*	τ_o	τ_p
2	0.02	0.80	0.1	0.2374	33	34
2	0.02	0.40	0.1	0.2417	145	148
3	0.01	0.10	1.0	0.6564	18	22
3	0.02	0.10	1.0	0.4812	52	51
3	0.01	0.05	1.0	0.4806	106	102
3	0.01	0.05	0.5	0.4793	218	203

6 Conclusions

A model has been presented with uncertainty in actions, group dynamics, payoffs, and state information. Learning automata achieve equilibrium in the particular cases examined with instantaneous information. This means that an agent successfully employs an automaton at each of the two levels. However, with delays in the system, the behaviors may exhibit damped or persistent oscillations and the onset of chaotic regimes.

The analysis yields the delay required to initiate persistent oscillations; unfortunately, the parameter settings that decrease the likelihood of instabilities also increase the likelihood that a suboptimal equilibrium will result. This illustrates the fundamental problem of seeking the optimum strategy without being misled by delayed information. However, the analysis is useful in that agents which communicate often enough to insure $\tau < \tau_2$ are guaranteed that persistent oscillations will not develop, thus insuring the stability of the system. This can have a strong impact on the performance of the system as stability is usually a prerequisite for good performance. In general, stability also is a measure of successful learning.

References

[Axelrod and Hamilton, 1981] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *Science*, 211:1390–1396, Mar. 1981.

[Billard and Pasquale, 1993a] E. A. Billard and J. C. Pasquale. Dynamic scope of control in decentralized job scheduling. In *Proc. IEEE Intl. Symp. Autonomous Decentralized Syst.*, pages 183–189, Mar. 1993.

[Billard and Pasquale, 1993b] E. A. Billard and J. C. Pasquale. Effects of delayed communication in dynamic group formation. *IEEE Trans. Syst., Man, Cybern.*, 23(5):1265–1275, Sept./Oct. 1993.

[Farmer, 1982] J. D. Farmer. Chaotic attractors of an infinite-dimensional dynamic system. *Physica D*, 4:366–393, 1982.

[Genesereth *et al.*, 1985] M. R. Genesereth, M. L. Ginsberg, and J. S. Rosenschein. Cooperation without communication. In *1985 Workshop on Distributed Artificial Intelligence*, pages 220–226, Sea Ranch, California, 1985.

[Glockner and Pasquale, 1993] A. Glockner and J. C. Pasquale. Coadaptive behavior in a simple distributed job scheduling system. *IEEE Trans. Syst., Man, Cybern.*, 23(3):902–907, May/June 1993.

[Gmytrasiewicz *et al.*, 1991a] P. J. Gmytrasiewicz, E. H. Durfee, and D. K. Wehe. Combining decision theory and hierarchical planning for a time-dependent robotic application. In *Proc. 7th IEEE Conf. on AI Applications*, pages 282–288, Feb. 1991.

[Gmytrasiewicz *et al.*, 1991b] P. J. Gmytrasiewicz, E. H. Durfee, and D. K. Wehe. A decision-theoretic approach to coordinating multiagent interactions. In *Proc. 12th Intl. Joint Conf. on AI*, pages 166–172, 1991.

[Gmytrasiewicz *et al.*, 1991c] P. J. Gmytrasiewicz, E. H. Durfee, and D. K. Wehe. The utility of communication in coordinating intelligent agents. In *Proc. 9th Natl. Conf. on AI*, pages 166–172, July 1991.

[Halpern and Moses, 1990] J. Y. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. *J. ACM*, 37(3):549–587, July 1990.

[Hogg and Huberman, 1991] T. Hogg and B. A. Huberman. Controlling chaos in distributed systems. *IEEE Trans. Syst., Man, Cybern.*, 21(6):1325–1332, Nov./Dec. 1991.

[Huberman and Hogg, 1988] B. A. Huberman and T. Hogg. The behavior of computational ecologies. In B. A. Huberman, editor, *The Ecology of Computation*. Elsevier Science Publishers, North-Holland, 1988.

[Kephart *et al.*, 1989] J. O. Kephart, T. Hogg, and B. A. Huberman. Dynamics of computational ecosystems. *Phys. Rev. A*, 40:404–421, July 1989.

[Lakshmivarahan and Narendra, 1982] S. Lakshmivarahan and K.S. Narendra. Learning algorithms for two-person zero-sum stochastic games with incomplete information:A unified approach. *SIAM J. Control and Optimization*, 20(4):541–552, July 1982.

[Narendra and Thathachar, 1989] K.S. Narendra and M.A.L. Thathachar. *Learning Automata: An Introduction*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

[Rosenschein and Genesereth, 1985] J. S. Rosenschein and M. R. Genesereth. Deals among rational agents. In *Proc. 9th Intl. Joint Conf. on AI*, pages 91–99, Aug. 1985.

[Wiener and Hale, 1992] J. Wiener and J. K. Hale, editors. *Ordinary and Delay Differential Equations*. John Wiley & Sons, New York, 1992.

Learning in Single- and Multi-Level Games with Delayed Information

Edward A. Billard
Faculty of Computer Science and Engineering
University of Aizu, 965-80 Japan
email: billard@u-aizu.ac.jp

Abstract

Distributed decision makers are modeled as players in a game with two levels. High level decisions concern the game environment and determine the willingness of the players to form a coalition (or group). Low level decisions involve the actions to be implemented within the chosen environment. Coalition and action strategies are determined by probability distributions which are updated using learning automata schemes. The payoffs are also probabilistic and there is uncertainty in the state vector since information is delayed. The goal is to reach equilibrium in both levels of decision making; the results show the conditions for instability, based on the age of information.

Keywords: coalitions, delay differential equations, delayed information, distributed decision-making, dynamical systems, group formation, learning automata, linear stability analysis, oscillations, stochastic games .

1 Introduction

Agents in a distributed system make decisions to optimize a performance metric or achieve a more abstract set of goals. These agents must typically consider working with other agents to cooperatively achieve the desired result. However, there is a high degree of uncertainty in these activities. First, the agent may not know the true state of the system as a result of delayed information. The delays may be due to inherent latencies in a network or the intermittent (or periodic) exchange of information. The agents make the best possible decisions with the information available [1]. Second, even with instantaneous information, there is uncertainty in the strategies employed by the other agents given the state vector. For example, an agent may not be certain that another agent is willing to cooperate or to what extent. Third, even with knowledge of the other strategies, there is uncertainty in the payoffs that result from the combined actions.

We present a model to capture these various uncertainties with distributed decision makers as players in a game with two levels. The high level concerns the game environment and determines the willingness of the players to form a coalition (or group). The low level involves the actions to be implemented within the chosen environment.

Both of these strategies are modeled using probability distributions with updates according to learning automata schemes [2]. This implies that learning is taking place on two levels and a constraint is that a player must make both decisions simultaneously, without knowledge of the other players' decisions at either level. In particular, a player knows whether it is willing to form a group but does not know the intentions of the other players. This implies that a player may select an action under the assumption of cooperative behavior but this action may result in suboptimal performance with the failure of group formation.

The payoffs in the games are stochastic, that is, there is a probability of gain or loss based upon the action set. Uncertainty in information is captured by the assumption that an average age of information exists in the system. The goal of the model is to capture decision making under uncertainty in various domains and to summarize uncertainty as probability distributions. The adaptive learning schemes easily model the uncertainty, permit expected value computations to determine beliefs, and have analytic solutions to complex dynamical behaviors. These schemes may also be considered as approximations to more complex reasoning schemes. We develop a dynamical equation to predict the behavior based on the parameter settings and apply linear stability analysis to predict the onset of persistent oscillations in the strategies.

The paper is organized as follows: Sec. 2 describes related work; Sec. 3 develops the model in stages, including the dynamical equation. Example simulations and associated predicted behavior are shown for single games and hiearchical games in Sec. 4 and Sec. 5, respectively. Linear stability analysis, in Sec. 6, shows the conditions for instability in single games and can be used as an approximation for instability in hierarchical games. Our conclusions are presented in Sec. 7.

2 Related Work

Our interests in distributed decision making [3] are closely related to computational ecosystems [4] [5] [6] and the evolution of cooperation [7]. The original description of computational ecologies [4] shows the dynamical equation based on simple gain functions with imperfect and delayed information. The resultant behavior can be categorized as stable, oscillatory (both damped and persistent), or chaotic (with possible bifurcations). The agreement between the dynamical equation and simulation is demonstrated in [6] and the existence of a general adaptive strategy to eliminate the instabilities is shown in [5] (although for instantaneous rewards).

In distributed computing systems, a high degree of physical decentralization [8] leads to aged information such that agents are not able to attain common knowledge [9]. The goal of agents in these systems is to make good decisions with the information available and, in particular, to make good decisions involving cooperation with other agents. Other research examines cooperation without communication [10].

Our approach is to examine learning mechanisms such as learning automata [2] in environments with delayed information. The basic research relevant to automata playing stochastic games (and the associated dynamics) is found in [11]. Our model extends this to delayed information and a hierarchy of games. The games in our model are intended to represent some underlying application (examples of such games can be found in [12] [1] [13]).

Learning automata are useful in distributed systems, an example is coadaptive behavior in a queueing system [14]. We have also examined learning automata in autonomous decentralized queueing systems [15] and in games [16]. (The current work subsumes [16] in that learning occurs on multiple levels.) We view the learning algorithms as generic in the sense that they capture incremental, or adaptive, learning. Other models capture the problem of cooperation in, for example, negotiated protocols [17].

Although increased levels of communication can reduce the age of information to the minimum latency, there is an associated cost in processing this information. For this reason, it is important to exchange the appropriate information. This can be done based on expected utility [1] with agents reaching equilibrium using recursive reasoning [12].

3 The Model

The model is developed in four stages: 1) the basic algorithm for a learning automaton [2], 2) the algorithm applied to the strategies of two players in a game, 3) the algorithm applied again to the strategies of selecting between two games, and 4) the delay in state information. In addition, the delay differential equation is shown.

Step 1: One Automaton - Two Strategies

Let $p(t)$ and $\bar{p}(t)$ be the probability of selecting strategy 1 and strategy 2, respectively, at time t. The probability is incremented or decremented for the next time step by

$$\Delta p = \theta \cdot \begin{cases} +\beta\bar{p} & \text{if reward on strategy 1} \\ -\beta p & \text{if reward on strategy 2} \\ -\alpha p & \text{if penalty on strategy 1} \\ +\alpha\bar{p} & \text{if penalty on strategy 2} \end{cases} . \qquad (1)$$

The extent of the incremental change in the mixed strategy is determined by the three constants: β is the reward parameter, α is the penalty parameter, and θ is the step size parameter. It is assumed that $0 < \alpha < \beta < 1$ and $0 < \theta \leq 1$. Although θ can be incorporated into α and β, it is convenient to extract this term for simulation and analysis results.

Step 2: Two Players - Two Strategies

We define two players $k, l \in \{1,2\}$ in a game $\mathbf{D} = (\mathbf{D}^1, \mathbf{D}^2)$, where $\mathbf{D}^k$ represents a stochastic payoff matrix for player k [11] [2] and corresponds to an underlying application. Each player chooses a strategy $i, j \in \{1,2\}$, respectively, and the game is played in stages with element d_{ij}^k of $\mathbf{D}^k$ being the probability of a unit gain for player k based upon the strategy pair (i, j). With probability 1-d_{ij}^k, player k receives a unit loss. This differs from games with deterministic payoffs as there is uncertainty in the result based upon the strategy pair. In the model, the game payoffs are the expected difference in gain and loss, $g_{ij}^k = 2d_{ij}^k - 1$, which scales to the interval $[-1, +1]$. The bi-matrix $\mathbf{D}$ is a nonzero-sum game such that both players may receive a unit gain (or unit loss), that is, d_{ij}^1 does not necessarily equal 1-d_{ij}^2.

The decisions are made using randomization and, as such, both players are uncertain as to the pure strategy that will be employed by the other player. Let $\mathbf{p} = (p_1, p_2)$ be the state vector where p_k is the probability that player k will select strategy 1 and $\bar{p}_k$ is the probability of strategy 2. Each player employs an automaton to update the probabilities for the next stage where a unit gain is a reward and a unit loss is a penalty.

The following closely parallels the derivation in [11] except that we include nonzero-sum games, delayed information, and a more general notation that permits learning in a hierarchy of games.

Let $\delta z(t) = z(t+1) - z(t)$. The expected change in the probability vector can be deduced from Eqn. 1. For example, with probability p_1, player 1 will select strategy 1. If the player receives a reward, then p_1 will increment by $\theta\beta\bar{p}_1$. Following this reasoning for all possibilities:

$$E[\delta \mathbf{p}(t) | \mathbf{p}(t) = \mathbf{p}] = \theta \mathbf{W}(\mathbf{p}), \tag{2}$$

where

$$W_k(\mathbf{p}) = \beta p_k \bar{p}_k [C_1^k(\mathbf{p}) - C_2^k(\mathbf{p})] + \alpha[\bar{p}_k^2 \bar{C}_2^k(\mathbf{p}) - p_k^2 \bar{C}_1^k(\mathbf{p})] \tag{3}$$

and $C_i^k(\mathbf{p})$ is the probability that player k receives a reward for strategy i. This is determined as follows. Let $\mathbf{p}_k = (p_k \; \bar{p}_k)$ be the probability vector for player k. The expected game payoff, or value of the game, for player k is

$$\eta_k(\mathbf{p}) = \mathbf{p}_1 \mathbf{D}^k \mathbf{p}_2^T, \tag{4}$$

where $\mathbf{p}_2^T$ is the transpose of $\mathbf{p}_2$. Now, $C_i^k(\mathbf{p}) = \eta_k(\mathbf{q})$ where $\mathbf{q} = \mathbf{p}$ but with the kth element replaced by 2-i. For example, if player 1 selects strategy 1, then the expected payoff is $p_2 d_{11}^1 + \bar{p}_2 d_{12}^1$.

We recast the difference equation as a differential equation as this closely captures the behavior for the typical parameter settings, i.e. small θ. Therefore,

$$\frac{d\mathbf{p}}{dt} = \theta\mathbf{W}(\mathbf{p}). \quad (5)$$

The equilibrium solution is $\mathbf{p}^*$ where $\mathbf{W}(\mathbf{p}^*) = 0$. Note that the values of the learning parameters affect the equilibrium solution, that is, $\mathbf{p}^*=f(\alpha, \beta, \mathbf{D})$.

Step 3: Four Players - Two Games

We introduce the concept of multi-level games to capture the notion of cooperation in group dynamics, see Fig. 1. An agent consists of two subcomponents, or players, each of which is modeled as a learning automaton. One player within each agent makes a preference decision between two game bi-matrices **A**, the *non-default* game matrix, and **B**, the *default* game matrix. Game **B** represents the underlying environment when the agents choose not to form a group. Typically, the payoffs will be lower but easier to achieve (in the sense of an equilibrium). Game **A** represents the environment when both agents agree to cooperate in a group with the expectation that better payoffs are available to both agents. However, to achieve these payoffs, the agents must successfully coordinate their actions within the game, perhaps a more difficult task in this game than in **B**. This second activity, i.e. selecting an action strategy within the chosen game environment, is carried out by an additional player within each agent. If an agent is willing to play game **A**, there is uncertainty whether the other agent will agree and, hence, the player subcomponent may make poor action decisions. For example, player 1 may select action strategy 1 since it has a high expectation of success in game **A**, the agent's preferred matrix. If agent 2 forces the default game environment, strategy 1 may yield a very poor result. It is the uncertainty in coalition formation and the simultaneity of decision making that makes action decision making difficult.

We define the high level decisions (i.e. *which* game matrix) as group strategies and the low level decisions (i.e. which strategy *within* a game) as action strategies. The formal definition of the model is as follows.

The action strategies are determined as before (using p_1 and p_2). The group strategies are also made using randomization with p_3 the probability that player 3 (a subcomponent of agent 1) will prefer **A** over **B** (likewise, p_4 is the probability for player 4, a subcomponent of agent 2). The state vector is now $\mathbf{p} = (p_1, p_2, p_3, p_4)$. At the high level, each player uses an automaton to decide the game preference. At the low level, each

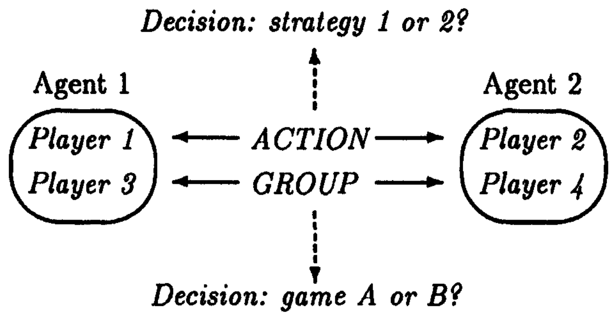


Figure 1: Multi-Level Decision Making by Agents' Components

player uses a different automaton to select a strategy. The action pair is determined at the same time as the group decision. The resultant action pair (i, j) is played in game $\mathbf{A}$ if, and only if, *both* agents prefer this game matrix. That is, the agents agree to form a coalition with probability $c = p_3 p_4$, the clustering parameter. Otherwise, the stochastic payoffs are determined by $\mathbf{B}$ with the agents operating in a non-coalition mode. The problem of apportioning credit to the different levels is avoided by assuming that both levels receive the same payoff, that is, both receive either a unit gain or unit loss.

An average game is induced based on the high level strategies:

$$\mathbf{D}^k = c \cdot \mathbf{A}^k + (1 - c) \cdot \mathbf{B}^k. \tag{6}$$

The dynamical equation is still Eqn. 5 but where $k \in \{1, 2, 3, 4\}$ and $\mathbf{D}^k = \mathbf{D}^{k-2}$ for $k \in \{3, 4\}$. Note that this equation enforces a strong interaction among the state variables. The low level strategies are dependent on the high level strategies for the expectation of the average game. They are also dependent on each other via the stochastic payoffs based on action pairs. The high level strategies are dependent on the low level strategies since the reward (or penalty) is derived in the same way. The potential exists for different learning rates at different levels but, in this study, both rates are identically θ.

Step 4: Delayed Information

Since agents are physically distributed, the information available to an agent is delayed. The state vector $\mathbf{p}$ describes the probabilities of decisions at both the high and low level and, in our model, is subject to aged information. That is, the agents must make the best decisions possible given an aged view of the likelihood of the other agent's decisions.

Let τ be the average delay in information, representing the overall effect of latency within the distributed system. For example, latency is increased by periodic broadcasts of information or by the inherent delays within network hardware and software. The latency is a fundamental cause of uncertainty.

Consider a probability $p_k(t)$. We define an aged view of this probability as $p_k^\tau = p_k(t\text{-}\tau)$ and $p_k(t) = p_k(0)$ for $t < 0$. Agent k knows with certainty the probability of its low and high strategies, p_k and p_{k+2}, respectively, and has an aged view of the other two probabilities. From the subcomponents point of view, let $\mathbf{p}^k$ be player k's view of the state vector, that is, $\mathbf{p}^1 = \mathbf{p}^3 = (p_1, p_2^\tau, p_3, p_4^\tau)$ and $\mathbf{p}^2 = \mathbf{p}^4 = (p_1^\tau, p_2, p_3^\tau, p_4)$.

In terms of the rules of the game, the preceding implies that a local module, or score keeper, provides a unit gain or loss based on the decisions of the local agent and the aged probabilities of the distant agent. For example, the local module for agent 1 determines the outcome based on the agent's pure strategy i and chance, but where chance is now determined by $p_2^\tau d_{i1} + \bar{p}_2^\tau d_{i2}$ (and the average game element d_{ij} is also based on aged information).

Now, Eqn. 5 may be applied using $W_k(\mathbf{p}^k)$ instead of the instantaneous vector $\mathbf{p}$. Formally, Eqn. 5 is a nonlinear delay differential equation [18].

4 Experiments with Single Games

In this section, we show the results of the dynamical equation and simulation for single games, that is, the agents do not have a choice between two games. Instead, the agents act as players within a single game, thus allowing the examination of elementary behaviors.

The following games are considered:

$$\mathbf{D}_1 = \begin{bmatrix} .6, .4 & .8, .2 \\ .35, .65 & .9, .1 \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} .75, 1 & .5, .25 \\ 1, .5 & .25, .75 \end{bmatrix}, \mathbf{D}_3 = \begin{bmatrix} .4, .4 & 1, 1 \\ 1, 1 & 0, 0 \end{bmatrix},$$

where $\mathbf{D}_1$ is a zero-sum game with a mixed strategy equilibrium [11], $\mathbf{D}_2$ is a nonzero-sum game with a mixed strategy equilibrium, and $\mathbf{D}_3$ is a nonzero-sum game with two pure strategy equilibria.

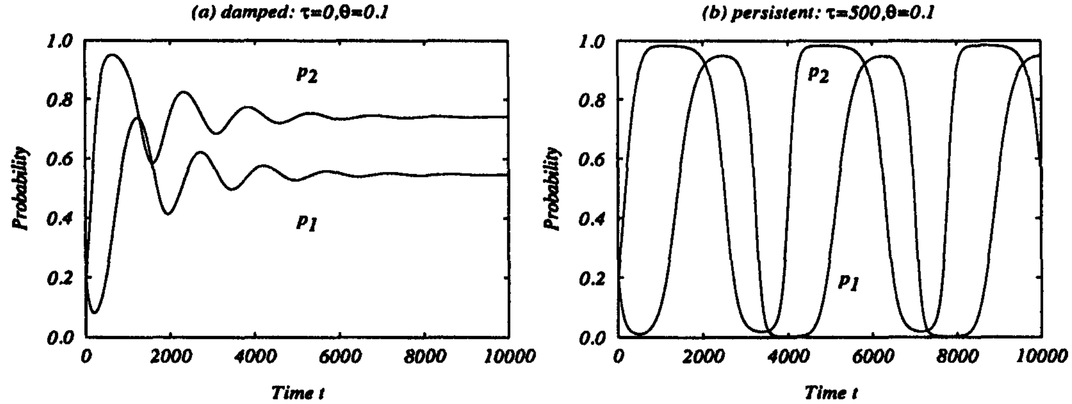


Figure 2: Theoretical Oscillations at Low and High Delay

Two examples of the behavior of the players in $\mathbf{D}_1$, based on the above dynamical equation, are shown in Fig. 2 for a total number of stages $n = 10000$ (α=0.005, β=0.2, θ=0.1). Figure 2a shows that, with no delay, damped oscillations reach an equilibrium that is very close to the optimum (as reported in [11]). However, persistent (and high amplitude) oscillations are observed with high delay, Fig. 2b.

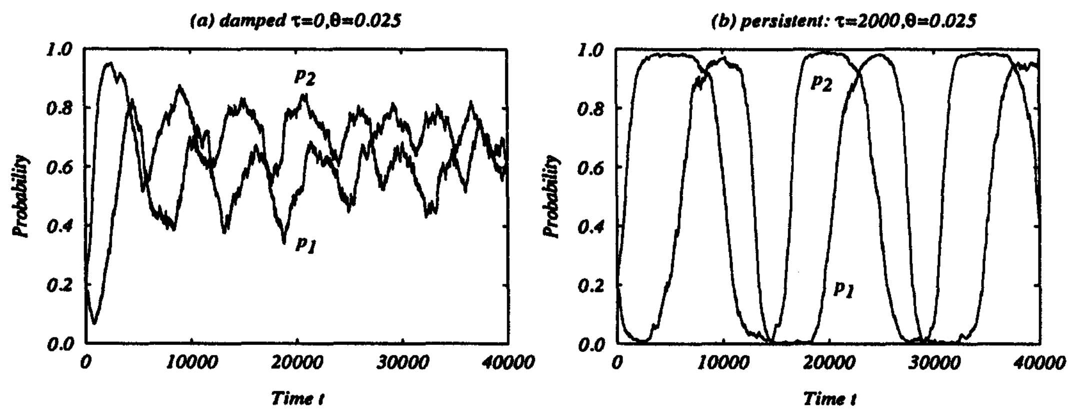


Figure 3: Simulation at Low and High Delay

Figure 3 shows the results of single simulation runs for the same experiment. Note that the relative delay is the same in both Fig. 3b and Fig. 2b, i.e. $\theta\tau$=50, though the individual parameters differ by a factor of 4. Small step sizes are necessary to yield accurate simulation runs.

In Fig. 3a, the oscillations do not show damped behavior but, instead, exhibit rough oscillations followed by irregular patterns. The high delay case, Fig. 3b, agrees closely with its counterpart, Fig. 2b. Although the simulation results show an approximate agreement with the results from the dynamical equation, multiple runs cannot be combined as small phase shifts eventually obliterate any oscillatory behavior (even though each individual run shows persistent oscillations [6]). An alternative is to correlate the data within each run and average the results [6].

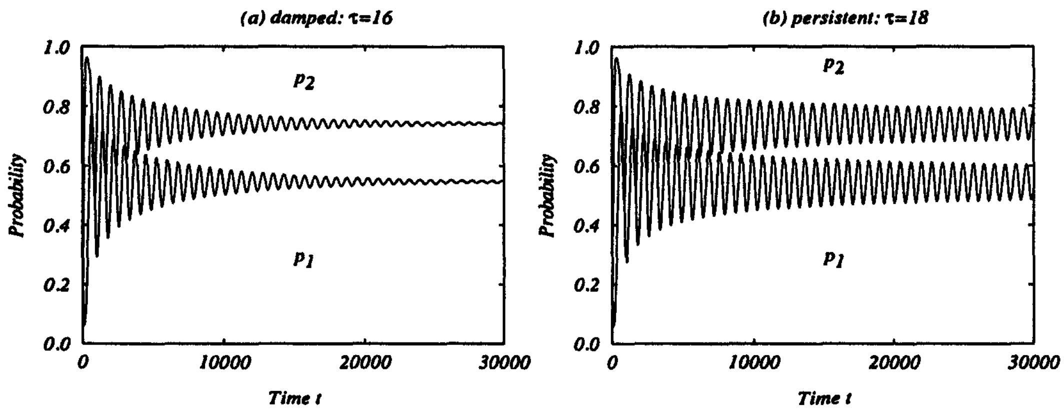


Figure 4: Oscillations Near Stability Boundary

Figure 4 shows the behavior near the boundary between damped and persistent oscillations (α=0.01, β=0.4, θ=0.1). In Fig. 4a, τ=16 and the mixed strategies exhibit damped oscillations which slowly settle into an equilibrium. However, in Fig. 4b, a small increase in τ (18) initiates persistent oscillations. We can say that the delay τ_2 required to initiate persistent oscillations is $16 < \tau_2 \leq 18$.

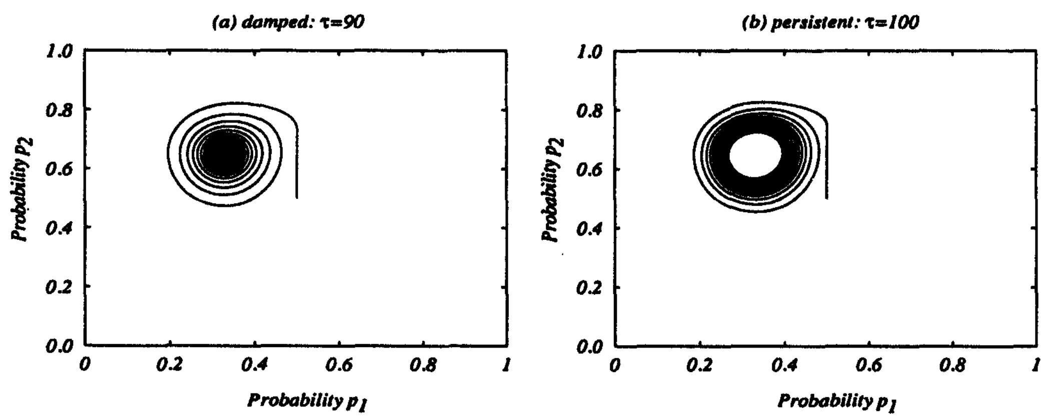


Figure 5: Phase-Plane Portrait Near Stability Boundary

Figure 5 shows phase-plane portraits, rather than time-dependent behavior, for $\mathbf{D}_2$. In the damped case, Fig. 5a, the oscillations reach an equilibrium such that the center of the spiral vanishes while the persistent oscillations, Fig. 5b, circulate about the attractor equilibrium. We observe that $90 < \tau_2 \leq 100$.

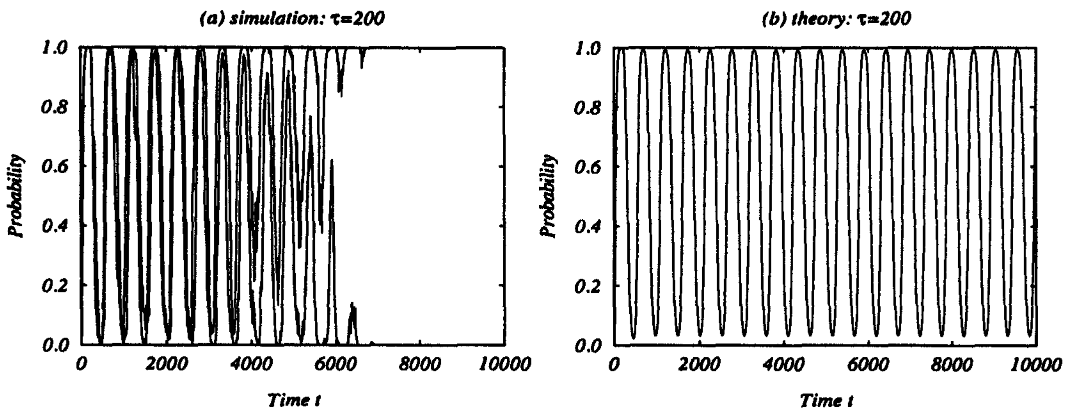


Figure 6: Randomization in Simulation Leads to Equilibrium

Figure 6b shows, theoretically, that the players oscillate synchronously between the two pure strategy equilibria in $\mathbf{D}_3$ for the given initialization. However, Fig. 6a shows that small phase shifts in the simulation eventually allow the players to reach an equilibrium in pure strategies.

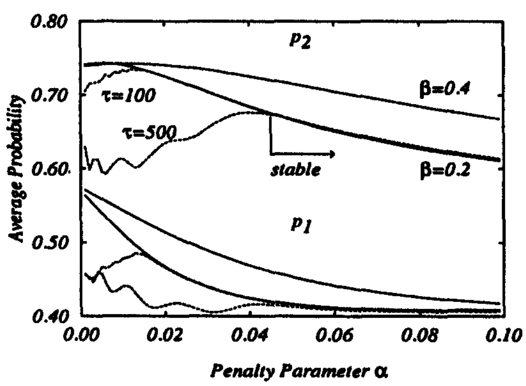


Figure 7: Effects of Learning and Delay on Average Strategy

The average mixed strategy (n=30000) is shown in Fig. 7 for two cases of β in $\mathbf{D}_1$. For τ=0 (solid lines), the average strategies approach the optima for low α, that is, the players learn the optimal strategies for sufficiently small parameter values (as in [11]). At a fixed α, a smaller value of β degrades the performance - it is best to have a large distance between the parameter values. Two examples of delay (dotted lines) are also shown for $\beta = 0.2$. As the delay increases, the performance decreases, however, there is a sufficient value of α for which the behavior approximates the instantaneous case. At the point where the delay and instantaneous cases agree, the persistent oscillations give way to damped oscillations, hence a stable, but suboptimal, equilibrium.

As the delay goes to zero, the only unstable parameter setting is α=0, which is the linear reward-inaction algorithm (L_{R-I}). This has been shown to oscillate due to non-negative real parts in the eigenvalues of the Hessian of the non-delayed version of (5) [11].

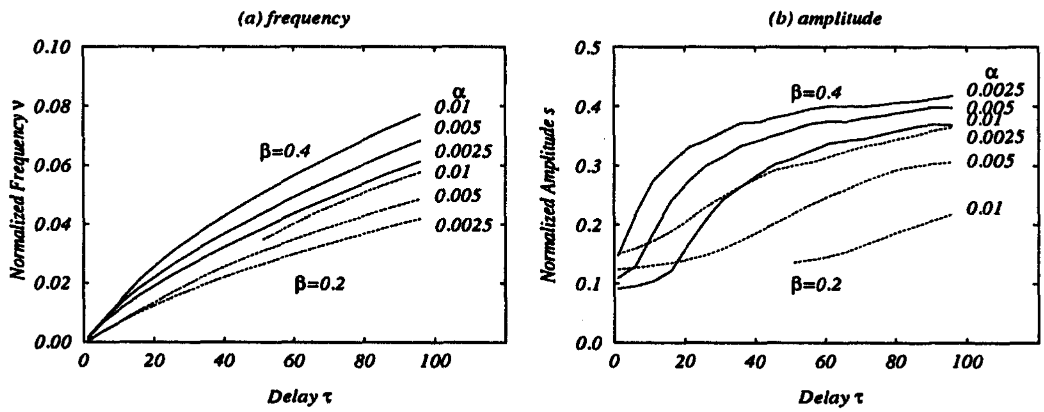


Figure 8: Effects of Learning Parameters and Delay

The characteristics of the oscillations, based on various parameter settings, are shown in Fig. 8 for $\mathbf{D}_1$. The normalized frequency of oscillation $\nu = \tau/T$, where T is the period of oscillation or time between successive maxima. The normalized frequency increases with delay τ and learning parameters α and β. Note that one particular instance (α=0.01, β=0.2) does not exhibit a sufficient number of maxima to determine the characteristics at low delay.

Fig. 8b shows the amplitude or extent of the oscillations in player 1's mixed strategy. The normalized amplitude s is the root mean square:

$$s = \sqrt{\Sigma_{t=1}^{n}(p_1(t) - P_{opt})^2/n},$$

which increases with delay τ and reward parameter β, but decreases with penalty parameter α. (Note that we measure the oscillations about the consistent value of the optimum rather than the average value for the particular parameter.) Figure 8 does not indicate the effect of the step size parameter on the oscillations but ν and s increase with θ.

5 Experiments with Hierarchical Games

In this section, we examine the implications of the full model, that is, the agents have a choice between game **A** or game **B**.

$\mathbf{A_1}$

1,1	0,0
0,0	0,0

$\mathbf{B_1}$

0,0	0,0
0,0	1,1

game 1

$\mathbf{A_2}$

.6,.4	.2,.8
.35,.65	.9,.1

$\mathbf{B_2}$

.4,.6	.8,.2
.65,.35	.1,.9

game 2

$\mathbf{A_3}$

.75,1	.5,.25
1,.5	.25,.75

$\mathbf{B_2}$

game 3

Figure 9: Example Games

Three games are considered with respect to learning behavior, see Fig. 9. The games are chosen to facilitate the illustration of key points and do not necessarily represent an underlying application. In game 1, the high level choice is between two game matrices, both with pure strategy equilibria of identical payoffs to both players. However, an opposite set of actions is required to achieve equilibrium. In game 2, the matrices are complements of each other and both are zero-sum game matrices with mixed strategy equilibria (the single game is from [11]). In game 3, one choice is a nonzero-sum game matrix with mixed strategy equilibrium and the other is the same default game matrix of game 2.

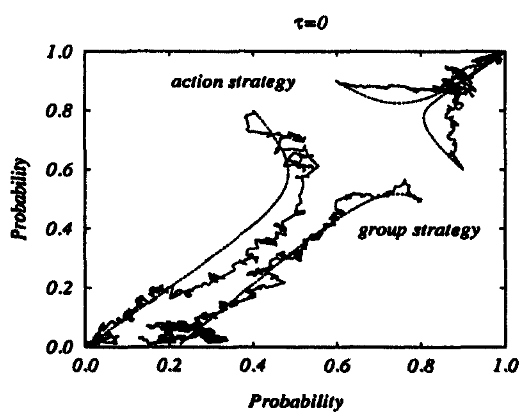


Figure 10: Phase-Plane Portrait of Game 1 with Two Pure Strategy Equilbria

Figure 10 shows the action and group strategies for game 1 in two experiments with different initialization (the delay in the system is zero.) The action strategies are plotted as p_2 versus p_1 and the group strategies as p_4 versus p_3. The initialization determines which of the two pure equilibria is "closest". The single runs roughly approximate the predicted behavior based on a numerical solution to Eqn. 5, that is, the players are able to reach an equilibrium in both levels. Note that the group strategy for the non-coalition equilibrium does not terminate at the origin. Instead, both strategies decrease at the same linear rate and whichever strategy reaches zero first (based on initialization) determines that a coalition does not form (i.e. $c{=}p_3p_4{=}0$). This linear behavior is due to the contrived nature of the game payoffs.

Figure 10 also shows that a large region of initialization is expected to result in the non-coalition equilibrium. For example, initialization $p_3{=}p_4{=}0.7$ is in the upper-right corner but is actually slightly biased to the non-coalition equilibrium ($c{=}0.49$).

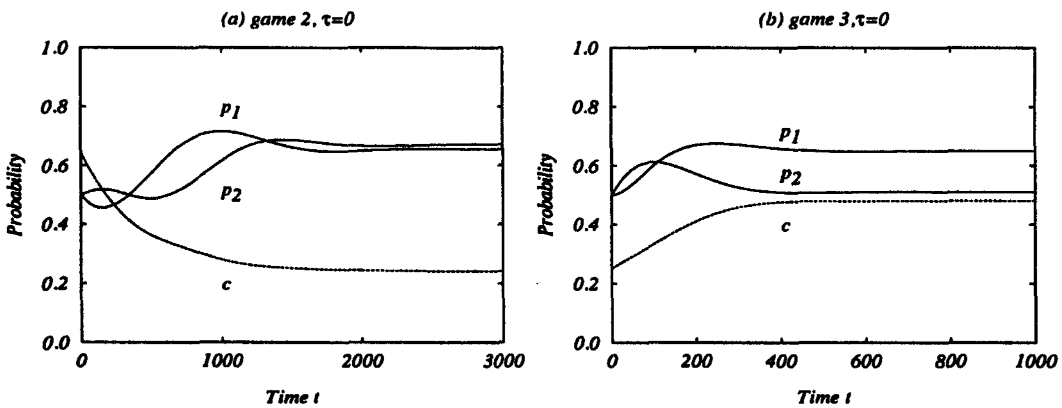


Figure 11: Equilibria in Action and Group Mixed Strategies

Figure 11 shows that players (theoretically) are able to reach mixed strategy equilibria (in both the group and action levels) for games 2 and 3 without delays. (Unless otherwise stated, $\alpha{=}0.02,\beta{=}0.4,\theta{=}0.01$ for game 2 and $\alpha{=}0.01,\beta{=}0.05,\theta{=}0.1$ for game 3). We now consider the effects of delays in the information exchanged in these two games.

Figure 12 shows both a single simulation run and the prediction of Eqn. 5 for game 2. In Fig. 12a, the amplitudes in the simulation are larger than predicted but would be reduced if a smaller parameter value was chosen. In both cases, the relative delay is the same, i.e. $\theta\tau = 15$, although the individual parameters in the two cases differ by an order of magnitude. For this reason, we may examine the theory with any value of θ, though we know that a small θ must be chosen to get an accurate simulation.

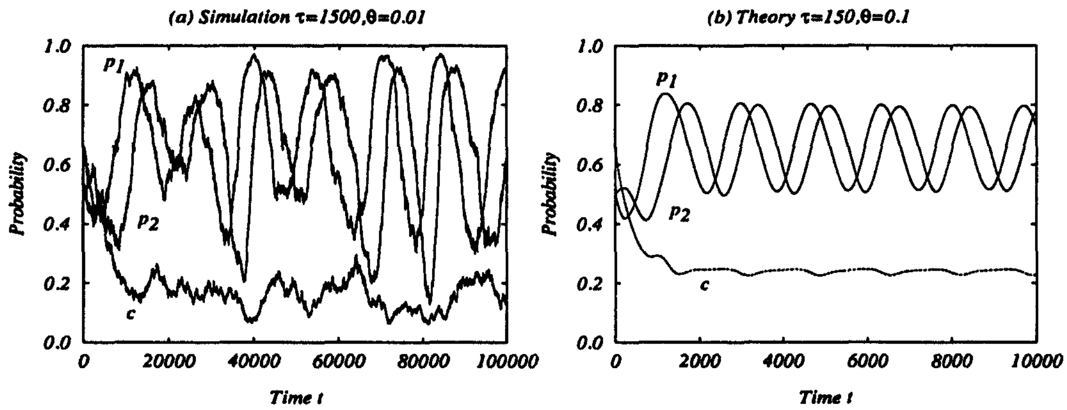


Figure 12: Oscillations in Action Strategies of Game 2

Note that persistent oscillations are predicted (for the action strategies) and we can say that the delay to initiate such oscillations, τ_2, must be less than or equal to 150 (for θ=0.1). At lower values of delay, the theoretical strategies exhibit damped oscillations, however, simulations do not typically show the theoretical damping but rather noise in the strategies.

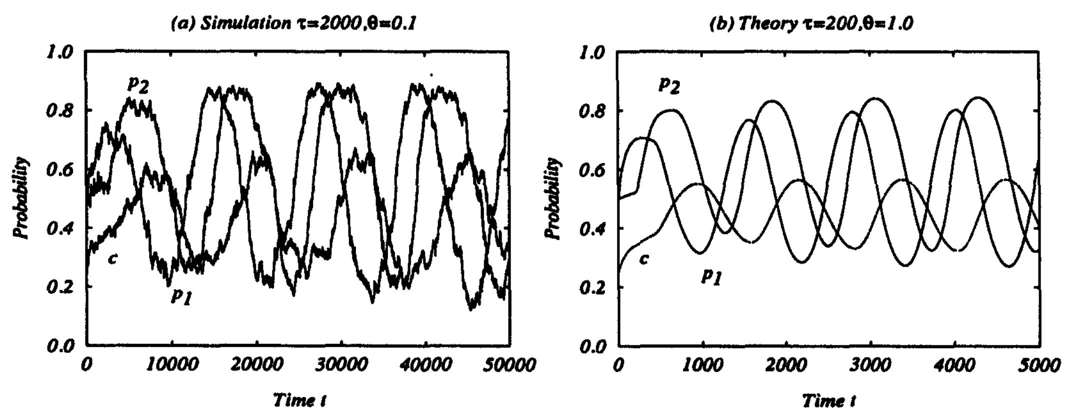


Figure 13: Oscillations in Action and Group Strategies of Game 3

Figure 12 showed that the players reach a rough equilibrium in the group strategies for game 2 but Fig. 13, for game 3, shows that the group strategies oscillate persistently. In this case, we can say that $\tau_2 <= 200$. (Other experiments with this game suggest that both the action and group strategies initiate oscillatory behavior at the same delay.) There is a rough approximation between theory and simulation, again with slightly higher amplitudes in simulation due to the step size parameter.

Figure 14 shows the predicted behavior of the action strategies for game 2 over a

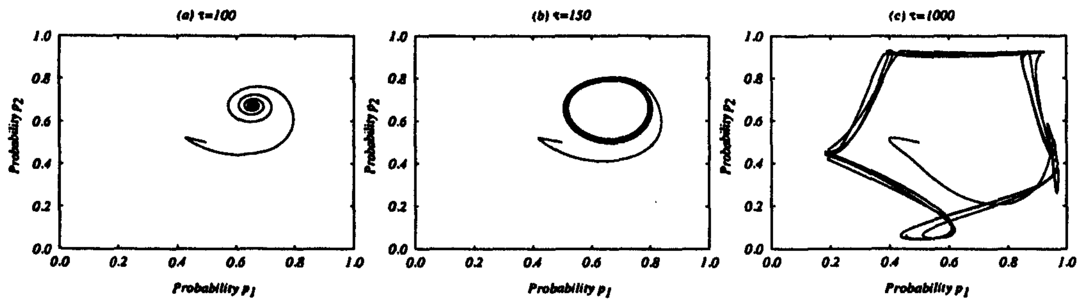


Figure 14: Phase-Plane Portraits of Action Strategies for Game 2 with Delay

small, medium, and large delay. These phase-plane portraits have the classic appearance of (a) damped oscillation, (b) persistent oscillation, and (c) chaos. The damped oscillations reach an equilibrium such that the center of the spiral vanishes while the persistent oscillations circulate about the attractor equilibrium. From this we can conclude that $100 < \tau_2 \leq 150$. The last case has multiple attractors and shifting behaviors. Note that Fig. 14b shows a longer run of the same experiment in Fig. 12b (but with a different type of plot) where the group strategies reach an equilibrium punctuated with slight periodic changes. Figure 15 shows the chaotic behavior of the group strategies under extremely high delay (corresponding to the action strategies of Fig. 14c).

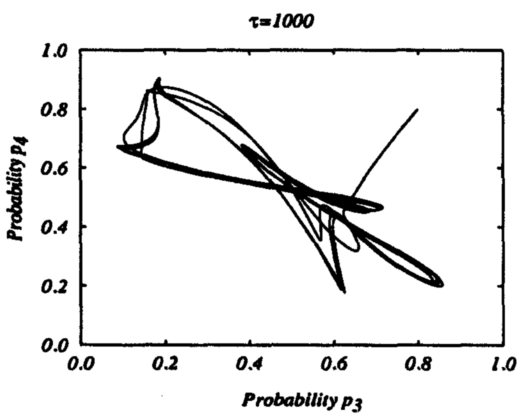


Figure 15: Phase-Plane Portrait of Group Strategies for Game 2 at High Delay

6 Analysis

The goal is to predict the onset of persistent oscillations using using linear stability analysis [19] which, for a stable attractor (as in our case), is equivalent to determining

Lynapunov exponents. We do this in two steps; first, we examine single games where the agents do not have a choice between games and, second, we show that a simple approximation for hierachical games works well.

6.1 Single Games

Let $k \in \{1,2\}$ and

$$X_k = \theta \frac{\partial W_k(\mathbf{p}^*)}{\partial p_k}, \quad Y_k = \theta \frac{\partial W_k(\mathbf{p}^*)}{\partial p_{3-k}^{\tau}}.$$

The partial derivatives are

$$\frac{\partial W_k(\mathbf{p})}{\partial p_k} = \beta(1-2p_k)[C_1^k(\mathbf{p}) - C_2^k(\mathbf{p})] - 2\alpha[\bar{p}_k \bar{C}_2^k(\mathbf{p}) + p_k \bar{C}_1^k(\mathbf{p})] \tag{7}$$

and

$$\frac{\partial W_k(\mathbf{p})}{\partial p_l^{\tau}} = \beta p_k \bar{p}_k \left[\frac{\partial C_1^k(\mathbf{p})}{\partial p_l^{\tau}} - \frac{\partial C_2^k(\mathbf{p})}{\partial p_l^{\tau}}\right] + \alpha \left[p_k^2 \frac{\partial C_1^k(\mathbf{p})}{\partial p_l} - \bar{p}_k^2 \frac{\partial C_2^k(\mathbf{p})}{\partial p_l}\right], \tag{8}$$

where

$$\frac{\partial C_i^k(\mathbf{p})}{\partial p} = \begin{cases} d_{i1}^k - d_{i2}^k & \text{if } k=1 \\ d_{1i}^k - d_{2i}^k & \text{if } k=2 \end{cases}. \tag{9}$$

Again, the results are similar to [11] but with a different notation.

Linearizing in the neighborhood of the equilibrium $\mathbf{p}^*$ [6] [19] [20] determines the delay conditions to initiate persistent oscillations. The system of equations is

$$\frac{d\delta p_k}{dt} = X_k \delta p_k + Y_k \delta p_{3-k}^{\tau}, \tag{10}$$

and assuming solutions of the form $\delta p_k(t) = A_k e^{\lambda t}$,

$$\lambda A_k e^{\lambda t} = X_k A_k e^{\lambda t} + Y_k A_{3-k} e^{\lambda(t-\tau)}. \tag{11}$$

Eliminating common terms and rearranging,

$$\lambda - X_k = (A_{3-k}/A_k) Y_k e^{-\lambda\tau}, \tag{12}$$

and this system yields:

$$(\lambda - X_1)(\lambda - X_2) = Y_1 Y_2 e^{-2\lambda\tau}. \tag{13}$$

Let $\lambda = r + iw$. There are an infinite number of discrete solutions and those parameter settings that yield only negative real parts are stable [6] [19] (with perhaps damped, but not persistent, oscillations). That is, marginal stability occurs at $r = 0$. The stability boundary can be determined by substituting $\lambda = iw$ in (13), applying Euler's formula, and solving for the real and imaginary parts:

$$\cos(2w\tau) = (X - w^2)/Y, \quad (14)$$

$$\sin(2w\tau) = (X_1 + X_2)w/Y, \quad (15)$$

respectively, where $X = X_1X_2$ and $Y = Y_1Y_2$. Dividing (15) by (14),

$$\tan(2w\tau) = \chi = \frac{(X_1 + X_2)w}{X - w^2} \quad (16)$$

and the instability delay (sufficient to initiate persistent oscillations) is:

$$\tau = \tau_2 = \tan^{-1}(\chi)/2w, \quad (17)$$

where the inverse tangent takes its value in the interval $[0, \pi/2]$.

Adding the squares of (14) and (15),

$$u^2 + Bu + C = 0, \quad (18)$$

where

$$u = w^2,\ B = X_1^2 + X_2^2,\ C = X^2 + Y^2,$$

hence $w = \pm\sqrt{u}$. The single solution to the quadratic equation is

$$u = \frac{-B + \sqrt{B^2 - 4C}}{2}, \quad (19)$$

as the other solution fails to insure a real (the only type of solution) for w (note that $B > 0$).

Before analyzing some specific details of these equations, we consider the behavior at various settings of reward and penalty parameters for the zero-sum game. Fig. 16 shows the instability delay, $\tau_2 = f(\alpha, \beta, \theta = 0.01, \mathbf{D}_1)$, that results from three different types of calculations. First, individual data points represent the onset of oscillations (the *observed* instability delay τ_o) as determined by (2), the strategies are examined over long runs at incremental delays to determine which delay is sufficient. Second, the solid lines are the solution to the predicted behavior (17) assuming that the equilibrium is the game-theoretic solution $\mathbf{p}_{opt}$, a simple *a priori* calculation [11]. The predicted behavior fails (by 20-25%) as α increases, as to be expected from using the wrong equilibrium.

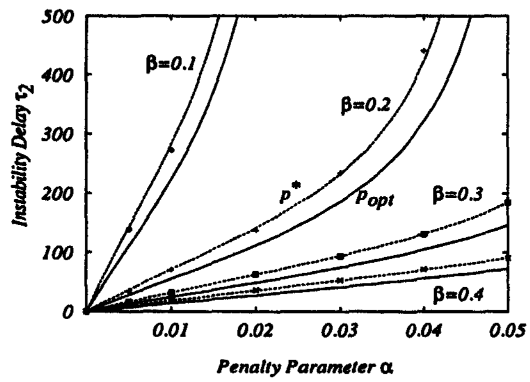


Figure 16: Stability Boundary

However, applying the expected equilibrium $\mathbf{p}^*$ to (17) yields the results in the dotted lines (the *predicted* instability delay τ_p). This approximates within 5% the observed results. In general, the delay required to initiate persistent oscillations increases with α and decreases with β. This is exactly *opposite* the conditions that insure an equilibrium that is close to the value of the game [11]. That is, parameters that are likely to lead to the optimal equilibrium in the non-delayed case are more likely to initiate instabilities in the delayed case.

The effect of θ is not shown but $\theta\tau$ is a measure of the relative delay in the system. In fact, the increase in the instability delay is proportionate to the decrease in θ, since the players do not make large steps in their decision-making process. However, a small θ also means that the players take longer to reach equilibrium.

We return now to details of (17); we examine zero-sum and nonzero-sum games with equilibria in mixed strategies and conclude the section with a discussion of pure strategy solutions. For mixed strategy solutions of zero-sum games, as in our example game, it can be verified that $X_1, X_2, Y_2 < 0$ and $Y_1 > 0$ (this has been shown, in particular for small α, in [11]). Consider the relevant interval $[-\pi/2, \pi/2]$ for the inverse tangent in (17). Since $X_1 + X_2 < 0$ and $Y = Y_1 Y_2 < 0$, then (15) implies that, if a solution exists, $\tau_2 \geq 0$ and the inverse tangent must be taken from the interval $[0, \pi/2]$. As the sign of the inverse tangent is opposite that of $X - w^2$, it is necessary to add π radians to χ in the case $X - w^2 > 0$.

It is apparent from (18) and (19) that $C = X^2 - Y^2 < 0$ is required to insure a real solution for w. For small values of α, this is true, as we can apply the results of [11]. As

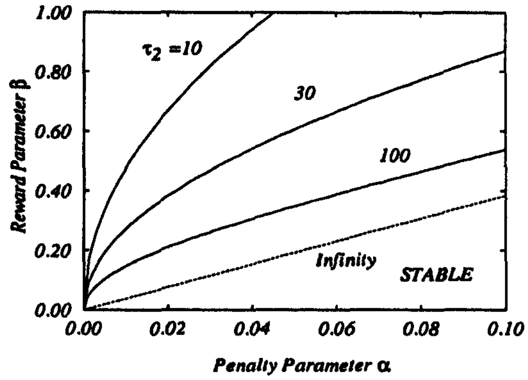


Figure 17: Stable Regions for High Delay

$\alpha \to 0$, $X_i \to 0$, but Y_i still has a β term. Therefore, $C \to -Y^2$, but, at large α, this is not the case. Consider the boundary C=0, that is, $w \to 0$. At this point, $\beta = f(\alpha)$, which can be determined from the partial derivatives. This may be done numerically at the equilibrium $\mathbf{p}^*$ or by assuming $\mathbf{p}_{opt}$ and applying the simplified partial derivatives [11]. The result, shown in Fig. 17, is $\beta = K\alpha$, where the constant K is based on the simplied case and is less than the value for $\mathbf{p}^*$. This boundary corresponds to infinite delay, that is, $\tau_2 = \infty$. Other delay cases are shown to approach this value and the plots are merely a reconstruction of the results in Fig. 16. In fact, Fig. 16, shows that the data for β=0.1 and β=0.2 are approaching asymptotes (this is the case for the other values of β, but the asymptotes are not included in the domain of the graph). As α increases beyond this asymptote, as defined in Fig. 17, the strategies are stable, albeit at a equilibrium that is distant from the optimum. This is confirmed by empirical evidence and suggested by the fact that (17) does not have a solution in this region.

We note that the linear stability analysis did not depend on the type of game, zero-sum or nonzero-sum. Also, our previous conclusions concerning (17) hold for, at least, small α as the Hessian in [11] is unaffected by the type of game. Consider the following nonzero-sum game

$$\mathbf{D}_2 = \begin{bmatrix} .75, 1 & .5, .25 \\ 1, .5 & .25, .75 \end{bmatrix}$$

which is a stochastic version of the mixed-strategy equilibrium game 70 in [21]. Table 1 shows close agreement between the predicted and observed values of instability delay for several parameter settings. As before, we can conclude that the delay required to initiate persistent oscillations increases with decreasing β, increasing α, and decreasing θ.

Table 1: Instability Delay: Predicted (τ_p) versus Observed (τ_o)

α	β	θ	τ_o	τ_p
0.01	0.50	0.1	15	15
0.01	0.25	0.1	56	54
0.1	0.5	0.1	100	99
0.1	0.5	0.05	206	198

Finally, we consider zero-sum games with equilibria in pure strategies. It can be verified that $C = X^2 - Y^2 > 0$ for all parameter settings, that is, a solution at the r=0 boundary does not exist and strategies are observed to be stable (in particular, as $\alpha \to 0$, $C \to X^2 > 0$ [11]). A corresponding graph to Fig. 17 would show that *all* parameter settings are safe, with respect to stability. For example, consider two zero-sum pure strategy equilibrium games [11]

$$\mathbf{D}_3 = \begin{bmatrix} .6,.4 & .8,.2 \\ .35,.65 & .9,.1 \end{bmatrix}, \qquad \mathbf{D}_4 = \begin{bmatrix} .7,.3 & .9,.1 \\ .6,.4 & .8,.2 \end{bmatrix},$$

where $\mathbf{D}_3$ has column dominance and $\mathbf{D}_4$ has both row and column dominance. In $\mathbf{D}_3$, player 2 prefers strategy 1 *regardless* of its view of the mixed strategy of player 1 and will incrementally move to the pure strategy solution. Whenever this preference is made known to player 1, it will also move toward the pure strategy solution. In $\mathbf{D}_4$, both players move toward the pure strategy solution regardless of the view of the other players' mixed strategy.

6.2 Hierarchical Games

In this section, an approximation is used to determine the amount of delay τ_2 required to initiate persistent oscillations when the agents have a choice between games. The technique involves linearizing in the neighborhood of the equilibrium $\mathbf{p}^*$ (as before) and the assumption that p_3 and p_4 are constant and equal to the equilibrium values in $\mathbf{p}^*$, that is, $\mathbf{c} = \mathbf{c}^*$. This implies that we ignore the partial derivatives with respect to these variables.

Table 2: Instability Delay: Predicted (τ_p) versus Observed (τ_o)

game	α	β	θ	c^*	τ_o	τ_p
2	0.02	0.80	0.1	0.2374	33	34
2	0.02	0.40	0.1	0.2417	145	148
3	0.01	0.10	1.0	0.6564	18	22
3	0.02	0.10	1.0	0.4812	52	51
3	0.01	0.05	1.0	0.4806	106	102
3	0.01	0.05	0.5	0.4793	218	203

The results are shown in Table 2 with the cases from Figs. 12 and 13 included. There is close agreement between the observed and predicted values of τ_2 and we can draw the same three conclusions as before: τ_2 increases with increasing α, decreasing β, and decreasing θ. The data suggests that ignoring the partial derivatives with respect to p_3 and p_4 did not hinder the analytic prediction (even though these probabilities oscillated in game 3).

7 Conclusions

A model has been presented with uncertainty in actions, group dynamics, payoffs, and state information. Learning automata achieve equilibrium in the particular cases examined with instantaneous information. This means that an agent successfully employs an automaton at each of the two levels. However, with delays in the system, the behaviors may exhibit damped or persistent oscillations and the onset of chaotic regimes. The analysis yields the delay required to initiate persistent oscillations; unfortunately, the parameter settings that decrease the likelihood of instabilities also increase the likelihood that a suboptimal equilibrium will result. This illustrates the fundamental problem of seeking the optimum strategy without being misled by delayed information.

We have concentrated on the effects of delay on the stability of the system and, as such, have not addressed any specific performance metric. If the metric is the system gain (the combined payoff in the game), then delays do not affect the constant result in zero-sum games. However, we have observed a 60% loss in a nonzero-sum game, with results depending on the game configuration. Independent of the game-theoretic model, losses typically arise in general systems during times that strategies are not in equilibrium - this is the reason that persistent oscillations are significant.

References

[1] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "The utility of communication in coordinating intelligent agents," *Proc. 9th Natl. Conf. on AI*, pp. 166–172, July 1991.

[2] K. Narendra and M. Thathachar, *Learning Automata: An Introduction.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] R. Tenney and N. Sandell, "Strategies for distributed decisionmaking," *IEEE Trans. Syst., Man, Cybern.*, vol. 11, no. 8, pp. 527–538, Aug. 1981.

[4] B. Hubermann and T. Hogg, "The behavior of computational ecologies," in *The Ecology of Computation* (B. Hubermann, ed.), North-Holland: Elsevier Science Publishers, 1988.

[5] T. Hogg and B. Huberman, "Controlling chaos in distributed systems," *IEEE Trans. Syst., Man, Cybern.*, pp. 1325–1332, Nov. 1991.

[6] J. Kephart, T. Hogg, and B. Huberman, "Dynamics of computational ecosystems," *Phys. Rev.*, vol. 40A, pp. 404–421, July 1989.

[7] R. Axelrod and W. Hamilton, "The evolution of cooperation," *Science*, vol. 211, pp. 1390–1396, Mar. 1981.

[8] E. Jensen, "Decentralized executive control of computers," in *Proc. IEEE Int. Conf. Distr. Comput. Syst.*, pp. 31–35, Oct. 1982.

[9] J. Halpern and Y. Moses, "Knowledge and common knowledge in a distributed environment," *J. ACM*, vol. 37, pp. 549–587, July 1990.

[10] M. Genesereth, M. Ginsberg, and J. Rosenschein, "Cooperation without communication," in *1985 Workshop on Distributed Artificial Intelligence*, (Sea Ranch, California), pp. 220–226, 1985.

[11] S. Lakshmivarahan and K. Narendra, "Learning algorithms for two-person zero-sum stochastic games with incomplete information:A unified approach," *SIAM J. Control and Optimization*, vol. 20, no. 4, pp. 541–552, July 1982.

[12] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "A decision-theoretic approach to coordinating multiagent interactions," *Proc. 12th Intl. Joint Conf. on AI*, pp. 166–172, 1991.

[13] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "Combining decision theory and hierarchical planning for a time-dependent robotic application," *Proc. 7th IEEE Conf. on AI Applications*, pp. 282–288, Feb. 1991.

[14] A. Glockner and J. Pasquale, "Coadaptive behavior in a simple distributed job scheduling system," *IEEE Trans. Syst., Man, Cybern.*, pp. 902–907, May/June 1993.

[15] E. Billard and J. Pasquale, "Dynamic scope of control in decentralized job scheduling," in *Proc. IEEE Int. Symp. Autonomous Decentralized Syst.*, pp. 183–189, Mar. 1993.

[16] E. Billard and J. Pasquale, "Effects of delayed communication in dynamic group formation," *IEEE Trans. Syst., Man, Cybern.*, no. 5, pp. 1265–1275, Sept./Oct., 1993.

[17] J. Rosenschein and M. Genesereth, "Deals among rational agents," in *Proc. 9th Intl. Joint Conf. on AI*, pp. 91–99, Aug. 1985.

[18] J. Wiener and J. Hale, eds., *Ordinary and Delay Differential Equations.* New York: John Wiley & Sons, 1992.

[19] J. Farmer, "Chaotic attractors of an infinite-dimensional dynamic system," *Physica*, vol. 4D, pp. 366–393, 1982.

[20] I. Györi and G. Ladas, *Oscillation Theory of Delay Differential Equations with Applications.* Oxford University Press, 1991.

[21] A. Rapoport and M. Guyer, "A taxonomy of 2 x 2 games," *General Systems: Yearbook of the Society for General Systems Research*, vol. 13, pp. 197–210, 1966.

Stability of Adaptive Search in Multi-Level Games under Delayed Information

Edward A. Billard
Faculty of Computer Science and Engineering
University of Aizu, 965-80 Japan

Abstract

Distributed decision makers are modeled as players in a game with two levels. High level decisions concern the game environment and determine the willingness of the players to form a coalition (or group). Low level decisions involve the actions to be implemented within the chosen environment. Coalition and action strategies are determined by probability distributions which are updated using learning automata schemes. The payoffs are also probabilistic and there is uncertainty in the state vector since information is delayed. The goal is to reach equilibrium in both levels of decision making. The results show the conditions for instability, based on the age of information, and a trade-off between optimality and stability.

1 Introduction

Agents in a distributed system make decisions to optimize a performance metric or achieve a more abstract set of goals. These agents must typically consider working with other agents to cooperatively achieve the desired result. However, there is a high degree of uncertainty in these activities. First, the agent may not know the true state of the system as a result of delayed information. The delays may be due to inherent latencies in a network or the intermittent (or periodic) exchange of information. The agents make the best possible decisions with the information available [1]. Second, even with instantaneous information, there is uncertainty in the strategies employed by the other agents given the state vector. For example, an agent may not be certain that another agent is willing to cooperate or to what extent. Third, even with knowledge of the other strategies, there is uncertainty in the payoffs or gain that result from the combined actions.

We are particularly interested in distributed environments where the choice of strategy is a statement regarding the type or level of group interaction. A model is presented to capture the three types of uncertainties involved in the search process for optimal group interaction. Distributed decision makers are modeled as players in a game with two levels. The high level concerns the game environment and determines the willingness of the players to form a coalition (or group). The low level involves the actions to be implemented within the chosen environment. The decision making at these two levels, which occurs repeatedly, may be encapsulated in single entities (e.g. a player) but also may be found in diverse entities in larger organizations.

Both of these strategies are modeled using probability distributions with updates according to learning automata schemes [2]. This implies that learning is taking place on two levels, we call these multi-level or hierarchical games. A constraint is that a player must make both decisions simultaneously, without knowledge of the other players' decisions at either level. In particular, a player knows whether it is willing to form a group but does not know the intentions of the other players. This implies that a player may select an action under the assumption of cooperative behavior but this action may result in suboptimal performance with the failure of group formation.

The adaptive learning schemes easily model the uncertainty, permit expected value computations to determine beliefs, and have analytic solutions to complex dynamical behaviors. These schemes may also be considered as approximations to more complex reasoning schemes. We develop a dynamical equation to predict the behavior based on the parameter settings and apply linear stability analysis to predict the onset of persistent oscillations in the strategies. Without delays, the agents are able to reach equilibria in both levels of decision making. With delays, the parameter settings that yield solutions closer to the optimum are the same settings that are more likely to lead to delay-induced instabilities; there is a trade-off between optimality and stability.

The paper is organized as follows: Section 2 describes related work; Section 3 develops the model in stages, including the dynamical equation. Example simulations and associated predicted behavior are shown for single games and hierarchical games in Section 4. Linear stability analysis, in Section 5 and the appendix, shows the conditions for instability in single games and can be used with an approximation to determine instability in hierarchical games. A large system of agents applying a generalized group decision algorithm is examined in Section 6. Our conclusions are presented in Section 7.

2 Related Work

Our interests in distributed decision making are closely related to computational ecosystems [3], [4] and the evolution of cooperation [5]. The resultant behavior in a computational ecosystem can be categorized as stable, oscillatory (both damped and persistent), or chaotic (with possible bifurcations). The agreement between the dynamical equation and simulation is demonstrated in [4] and the existence of a general adaptive strategy to eliminate the instabilities is shown in [3].

In distributed computing systems, a high degree of physical decentralization [6] leads to aged information such that agents are not able to attain common knowledge [7]. The goal of agents in these systems is to make good decisions with the information available and, in particular, to make good decisions involving cooperation with other agents.

Although increased levels of communication can reduce the age of information, there is an associated cost in processing this information. For this reason, it is important to exchange the appropriate information. This can be done based on expected utility [1] with agents reaching equilibrium using recursive reasoning [8].

Learning automata [2] are useful in distributed systems, an example is coadaptive behavior in a queueing system [9]. We have also examined learning automata in autonomous decentralized queueing systems [10] and in games [11], [12]. We view the learning algorithms as generic in the sense that they capture incremental, or adaptive, learning.

The basic research relevant to automata playing stochastic games (and the associated dynamics) is found in [13]. The games in the model are intended to represent some underlying application (examples of such games can be found in [8], [1], [14]). In this paper, our particular contribution in the context of learning automata is as follows. The original work [13] shows the differential equation, along with the partial derivatives, and that the resultant Hessian is stable. The model does not consider either delays or adaptation in multiple levels of games, the extensions in this study. The linear stability analysis, applied in the appendix, uses the partial derivatives in the context of a delay differential equation. We note that the analysis is of a standard form, as found in analog neural networks with delays [15], [16] and chaotic systems [4], [17]. Some of the original work in delay differential equations can be found in [18]. In many of these systems, the analysis is applied to simple linear equations. To our knowledge, this is the first time it has been applied to the delayed version of the nonlinear equations of learning automata, applied in multiple levels.

3 The Model

The salient features of the model are two players which make a high level choice between two games and, simultaneously, must try to make a good low level choice within the chosen game structure. Both decision mechanisms are modeled with learning automata subject to aged information regarding the probabilistic strategies of the other player. The goal is to capture the uncertainties present in a distributed system where an agent, or player, has a choice with respect to the best level of interaction with other agents. This also models discrete agents in a large organization, some of which are responsible for decisions involving group interactions, while others at a lower level, participate in other decisions.

The model is developed in four stages: 1) the basic algorithm for a learning automaton [2], 2) the algorithm applied to the strategies of two players in a game, 3) the algorithm applied again to the strategies of selecting between two games, and 4) the delay in state information. The first two steps follow from [13] and are provided to satisfy completeness and a new notation to accommodate the subsequent steps.

Step 1: One Automaton - Two Strategies

Let $p(t)$ and $\bar{p}(t)$ be the probability of selecting strategy 1 and strategy 2, respectively, at time t. The probability is incremented or decremented for the next time step by

$$\Delta p = \theta \cdot \begin{cases} +\beta\bar{p} & \text{if reward on strategy 1} \\ -\beta p & \text{if reward on strategy 2} \\ -\alpha p & \text{if penalty on strategy 1} \\ +\alpha\bar{p} & \text{if penalty on strategy 2} \end{cases} . \quad (1)$$

The extent of the incremental change in the mixed strategy is determined by the three constants: β is the reward parameter, α is the penalty parameter, and θ is the step size parameter. It is assumed that $0 < \alpha < \beta < 1$ and $0 < \theta \leq 1$. Although θ can be incorporated into α and β, it is convenient to extract this term for simulation and analysis results.

Step 2: Two Players - Two Strategies

We define two players $k, l \in \{1,2\}$ in a game $\mathbf{D} = (\mathbf{D}^1, \mathbf{D}^2)$, where $\mathbf{D}^k$ represents a stochastic payoff matrix for player k [13], [2] and corresponds to an underlying application. Each player chooses a strategy $i, j \in \{1,2\}$, respectively, and the game is played in stages with element d_{ij}^k of $\mathbf{D}^k$ being the probability of a unit gain for player k based upon the strategy pair (i,j). With probability 1-d_{ij}^k, player k receives a unit loss. The bi-matrix $\mathbf{D}$ is a nonzero-sum game such that both players may receive a unit gain (or unit loss), that is, d_{ij}^1 does not necessarily equal 1-d_{ij}^2.

The decisions are made using randomization and, as such, both players are uncertain as to the pure strategy that will be employed by the other player. Let $\mathbf{p} = (p_1, p_2)$ be the state vector where p_k is the probability that player k will select strategy 1 and $\bar{p}_k$ is the probability of strategy 2. Each player employs an automaton to update the probabilities for the next stage where a unit gain is a reward and a unit loss is a penalty.

The expected change in the probability vector can be deduced from (1). For example, with probability p_1, player 1 will select strategy 1. If the player receives a reward, then p_1 will increment by $\theta\beta\bar{p}_1$. Following this reasoning for all possibilities:

$$E[\delta\mathbf{p}(t)|\mathbf{p}(t) = \mathbf{p}] = \theta\mathbf{W}(\mathbf{p}), \quad (2)$$

where

$$W_k(\mathbf{p}) = \beta p_k\bar{p}_k[C_1^k(\mathbf{p}) - C_2^k(\mathbf{p})] + \alpha[\bar{p}_k^2\bar{C}_2^k(\mathbf{p}) - p_k^2\bar{C}_1^k(\mathbf{p})] \quad (3)$$

and $C_i^k(\mathbf{p})$ is the probability that player k receives a reward for strategy i. This is determined as follows. Let $\mathbf{p}_k = (p_k\ \bar{p}_k)$ be the probability vector for player k. The expected game payoff, or value of the game, for player k is

$$\eta_k(\mathbf{p}) = \mathbf{p}_1 \mathbf{D}^k \mathbf{p}_2^T, \tag{4}$$

where $\mathbf{p}_2^T$ is the transpose of $\mathbf{p}_2$. Now, $C_i^k(\mathbf{p}) = \eta_k(\mathbf{q})$ where $\mathbf{q} = \mathbf{p}$ but with the kth element replaced by 2-i. For example, if player 1 selects strategy 1, then the expected payoff is $p_2 d_{11}^1 + \bar{p}_2 d_{12}^1$.

We recast the difference equation as a differential equation as this closely captures the behavior for the typical parameter settings (i.e., small θ, see Section 4 for the accuracy of this assumption). Therefore,

$$\frac{d\mathbf{p}}{dt} = \theta \mathbf{W}(\mathbf{p}). \tag{5}$$

The equilibrium solution is $\mathbf{p}^*$ where $\mathbf{W}(\mathbf{p}^*) = 0$. Note that the values of the learning parameters affect the equilibrium solution, that is, $\mathbf{p}^* = f(\alpha, \beta, \mathbf{D})$.

Step 3: Four Players - Two Games

We introduce the concept of multi-level games to capture the notion of cooperation in group dynamics, see Fig. 1. An agent consists of two subcomponents, or players, each of which is modeled as a learning automaton. One player within each agent repeatedly makes a preference decision between two game bi-matrices **A**, the *non-default* game matrix, and **B**, the *default* game matrix. Game **B** represents the underlying environment when the agents choose not to form a group. Typically, the payoffs will be lower but easier to achieve (in the sense of an equilibrium). Game **A** represents the environment when both agents agree to cooperate in a group with the expectation that better payoffs are available to both agents. However, to achieve these payoffs, the agents must successfully coordinate their actions within the game, perhaps a more difficult task in this game than in game **B**. This second activity, i.e., selecting an action strategy within the chosen game environment, is carried out by an additional player within each agent. If an agent is willing to play game **A**, there is uncertainty whether the other agent will agree and, hence, the player subcomponent may make poor action decisions. For example, player 1 may select action strategy 1 since it has a high expectation of success in game **A**, the agent's preferred matrix. If agent 2 forces the default game environment, strategy 1 may yield a very poor result. It is the uncertainty and simultaneity that makes for difficult decisions.

We define the high level decisions (i.e., *which* game matrix) as group strategies and the low level decisions (i.e., which strategy *within* a game) as action strategies. The formal definition of the model is as follows.

The action strategies are determined as before (using p_1 and p_2). The group strategies are also made using randomization with p_3 the probability that player 3 (a subcomponent of agent 1) will prefer **A** over **B** (likewise, p_4 is the probability for player 4, a subcomponent of agent 2). The state vector is now $\mathbf{p} = (p_1, p_2, p_3, p_4)$. At the high level,

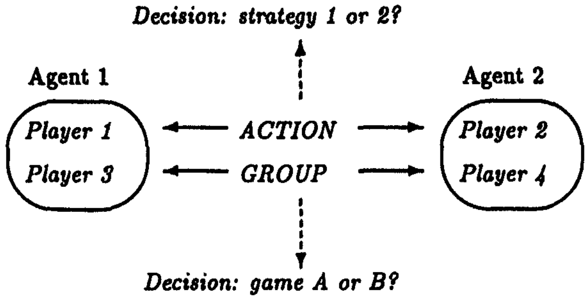


Figure 1: Multi-Level Decision Making by Agents' Components

each player uses an automaton to decide the game preference. At the low level, each player uses a different automaton to select a strategy. The action pair is determined at the same time as the group decision. The resultant action pair (i, j) is played in game **A** if, and only if, *both* agents prefer this game matrix. That is, the agents agree to form a coalition with probability $c = p_3 p_4$, the clustering parameter. Otherwise, the stochastic payoffs are determined by **B** with the agents operating in a non-coalition mode. The problem of apportioning credit to the different levels is avoided by assuming that both levels receive the same payoff, that is, both receive either a unit gain or unit loss.

An average game is induced based on the high level strategies:

$$\mathbf{D}^k = c \cdot \mathbf{A}^k + (1 - c) \cdot \mathbf{B}^k. \tag{6}$$

The dynamical equation is still (5) but where $k \in \{1, 2, 3, 4\}$ and $\mathbf{D}^k = \mathbf{D}^{k-2}$ for $k \in \{3, 4\}$. Note that this equation enforces a strong interaction among the state variables. The low level strategies are dependent on the high level strategies for the expectation of the average game. They are also dependent on each other via the stochastic payoffs based on action pairs. The high level strategies are dependent on the low level strategies since the reward (or penalty) is derived in the same way.

Note that in simulations the agents do not simply play the average game induced by the clustering parameter, as each subcomponent may be a discrete entity and/or be learning at a different rate. Each subcomponent of an agent makes a randomized decision, receives the same feedback, and adjusts its own probability (in this study, it is assumed that the learning rates are identically θ).

Step 4: Delayed Information

Since agents are physically distributed, the information available to an agent is delayed. The state vector **p** describes the probabilities of decisions at both high and low levels and, in our model, is subject to aged information. That is, the agents must make the best decisions possible given an aged view of the likelihood of the other agent's decisions. The agents cannot afford to communicate every decision, instead, the communication of the probabilistic strategies is intended to enhance the performance of the system. The strategies give an indication, at least for the near term, of the expected behavior of the players. Although it is not typical in game theory to exchange information regarding mixed strategies, we consider this quite appropriate for a distributed system, where randomization often appears. Also, the environments have a reasonably strong cooperative element (i.e., nonzero-sum games) and the purpose of this paper is to examine the behavior of agents attempting to determine optimal levels of group interaction.

Let τ be the average delay in information, representing the overall effect of latency within the distributed system. For example, latency is increased by periodic broadcasts of information or by the inherent delays within network hardware and software. The latency is a fundamental cause of uncertainty.

Consider a probability $p_k(t)$. The probability is communicated via update messages in an attempt to improve performance. We define an aged view of this probability as $p_k^\tau = p_k(t\text{-}\tau)$; note that $p_k(t) = p_k(0)$ for $t < 0$.

Agent k knows with certainty the probability of its low and high strategies, p_k and p_{k+2}, respectively, and has an aged view of the other two probabilities. From the subcomponent point of view, let $\mathbf{p}^k$ be player k's view of the state vector, that is, $\mathbf{p}^1 = \mathbf{p}^3 = (p_1, p_2^\tau, p_3, p_4^\tau)$ and $\mathbf{p}^2 = \mathbf{p}^4 = (p_1^\tau, p_2, p_3^\tau, p_4)$.

Now, (5) may be applied using $W_k(\mathbf{p}^k)$ instead of the instantaneous vector $\mathbf{p}$. Formally, (5) is a nonlinear delay differential equation [19].

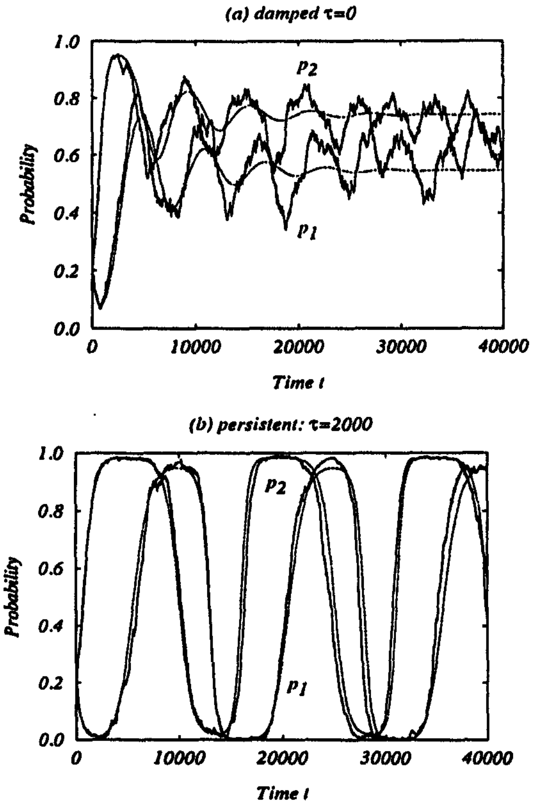


Figure 2: Simulation and Theory at Low and High Delay

4 Experiments

We examine the results of the dynamical equation and simulations for both single-level and multi-level games.

4.1 Single-Level Games

In this section, we examine the elementary behavior of agents which do not have a choice between two games. Instead, the agents act as players within a single game:

$$\mathbf{D}_1 = \begin{bmatrix} .6,.4 & .2,.8 \\ .35,.65 & .9,.1 \end{bmatrix},$$

where $\mathbf{D}_1$ is a zero-sum game with a mixed strategy equilibrium [13]. This game is used for motivational purposes only; the analysis in Section 5 works for any game with a mixed-strategy equilibrium.

Two examples of the behavior of the players in $\mathbf{D}_1$ are shown in Fig. 2 for a total number of stages $n = 40000$ (α=0.005, β=0.2). In Fig. 2(a), the dynamical equation predicts that, with no delay, damped oscillations reach an equilibrium that is very close to the optimum (as reported in [13]). However, persistent and high amplitude oscillations are observed with high delay, Fig. 2(b). In this case, the simulation results show close agreement with the theory since the step sizes are small ($\theta = 0.025$). In Fig. 2(a), the oscillations do not show damped behavior but, instead, exhibit rough oscillations followed by irregular patterns. There is a discrete boundary, based on delay, between damped and persistent oscillations, predicted in Section 5 and the appendix.

4.2 Multi-Level Games

In this section, we examine the implications of the full model, that is, the agents have a choice between game A or game B.

Three multi-level games are considered with respect to learning behavior, see Fig. 3. The games are chosen to facilitate the illustration of key points and do not necessarily represent an underlying application. The examples provide illustrations of the general behaviors, as analyzed in Section 5. In game 1, the high level choice is between two game matrices, both with pure strategy equilibria of identical payoffs to both players. However, an opposite set of actions is required to achieve equilibrium. In game 2, the matrices are complements of each other and both are zero-sum game matrices with mixed strategy equilibria (the single game is from [13]). In game 3, one choice is a nonzero-sum game matrix with mixed strategy equilibrium and the other is the same default game matrix of game 2.

Fig. 4 shows the action and group strategies for game 1 in two experiments with different initialization (the delay

A_1		B_1		A_2		B_2		A_3		B_2	
1,1	0,0	0,0	0,0	.6,.4	.2,.8	.4,.6	.8,.2	.75,1	.5,.25	.4,.6	.8,.2
0,0	0,0	0,0	1,1	.35,.65	.9,.1	.65,.35	.1,.9	1,.5	.25,.75	.65,.35	.1,.9
game 1				*game 2*				*game 3*			

Figure 3: Example Games

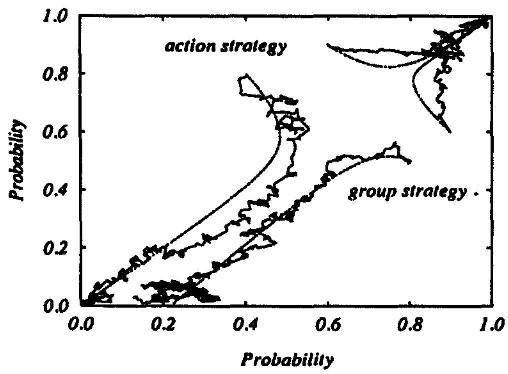


Figure 4: Phase-Plane Portrait of Game 1 with Two Pure Strategy Equilbria ($\tau = 0$)

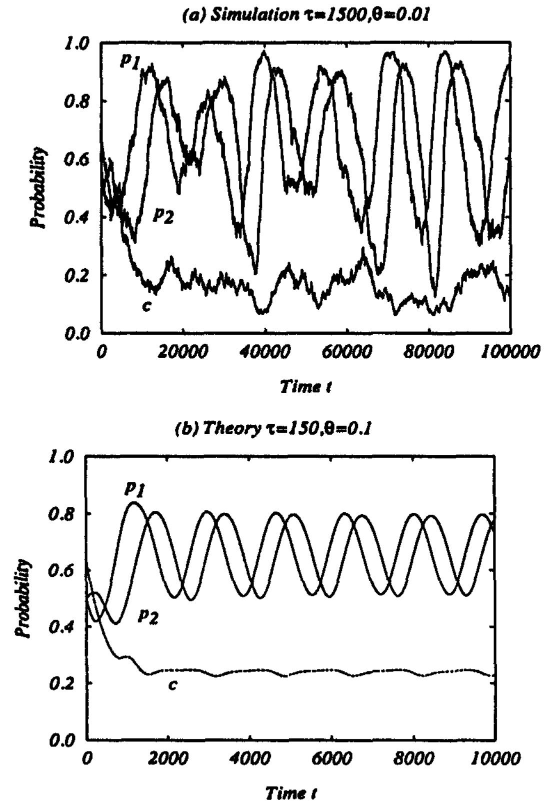


Figure 5: Oscillations in Action Strategies of Game 2

in the system is zero.) The action strategies are plotted as p_2 versus p_1 and the group strategies as p_4 versus p_3. The initialization determines which of the two pure equilibria is "closest". The single runs roughly approximate the predicted behavior based on a numerical solution to (5), that is, the players are able to reach an equilibrium in both levels. Note that the group strategy for the non-coalition equilibrium does not terminate at the origin. Instead, both strategies decrease at the same linear rate and whichever strategy reaches zero first (based on initialization) determines coalition failure (i.e., $c=p_3p_4=0$). Fig. 4 also shows that a large region of initialization is expected to result in the non-coalition equilibrium. For example, initialization $p_3=p_4=0.7$ is in the upper-right corner but is actually slightly biased to the non-coalition equilibrium (c=0.49).

The theory predicts that players are able to reach mixed strategy equilibria (in both the group and action levels) for games 2 and 3 without delays. This is an interesting result: the decision-making process using learning automata rules reaches equilibrium when applied at two levels. We now consider the effects of delays in the information exchanged in these two games.

Fig. 5 shows both a single simulation run and the prediction of (5) for game 2. (Unless otherwise stated, α=0.02, β=0.4, θ=0.01 for game 2 and α=0.01, β=0.05, θ=0.1 for game 3). In Fig. 5(a), the amplitudes in the simulation are larger than predicted but would be reduced if a smaller parameter value was chosen. In both cases, the relative delay is the same, i.e., $\theta\tau = 15$, although the individual parameters in the two cases differ by an order of magnitude. For this reason, we may examine the theory with any value of θ, though we know that a small θ must be chosen to get an accurate simulation. Note that persistent oscillations are predicted (for the action strategies) and we can say that the delay to initiate such oscillations $\tau_2 \leq 150$ (for θ=0.1).

Fig. 5 showed that the players reach a rough equilibrium in the group strategies for game 2 but Fig. 6, for game 3, shows that the group strategies oscillate persistently. In this case, we can say that $\tau_2 <= 200$. There is a rough approximation between theory and simulation, again with slightly higher amplitudes in simulation due to the step size parameter.

Fig. 7 shows the predicted behavior of the action strategies for game 2 for two delays near the stability boundary between damped and persistent oscillations. The damped

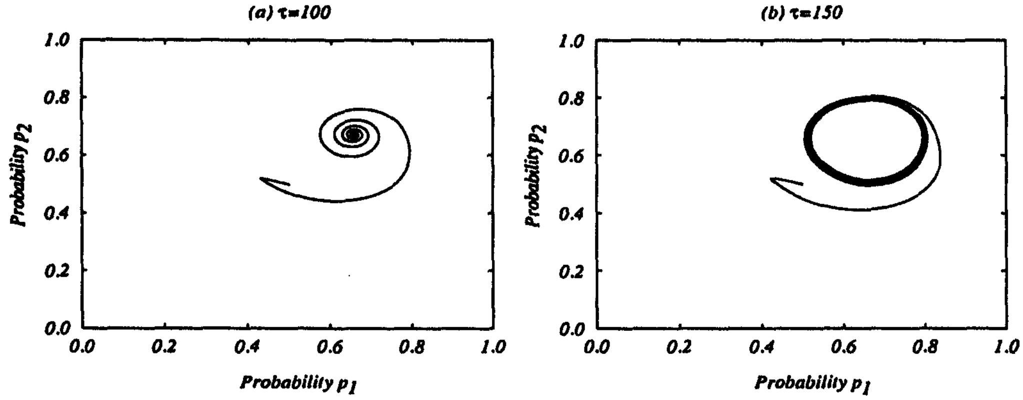


Figure 7: Phase-Plane Portraits of Action Strategies for Game 2 Near Stability Boundary

oscillations reach an equilibrium such that the center of the spiral vanishes and the same equilibrium serves as an attractor in the persistent oscillation case (i.e., limit cycle). From Fig. 7(a) and (b), we can conclude that $100 < \tau_2 \leq 150$.

Fig. 8 shows the onset of a chaotic attractor, with corresponding shifting behaviors, at very high delay. As noted in Section 3, there is a complex interaction between the two levels of learning: action strategies affect group strategies and vice versa. The high delay in the experiment induces the strategies to revisit a variety of potential equilibria, but with small shifts in the trajectory. Fig. 8(a) shows the specific behavior of the action strategies and Fig. 8(b) shows the behavior of the group strategies. Together, these two figures demonstrate, in four-dimensional space, the complex dynamics of learning at two levels under the circumstance of delayed information. This type of behavior is not found in single-level games with delayed information; it is only the multi-level decision-making process that initiates such behavior.

5 Analysis

In previous sections, we showed examples of single- and multi-level games. In this section, the analysis shows the general result. We do this in three steps; first, we demonstrate a trade-off between optimality and stability. Second, we predict the boundary between stability and instability for single-level games (see the appendix for details) and, last, we use an approximation to yield the boundary for multi-level games.

5.1 Optimality vs. Stability

There is a trade-off between optimality and stability; that is, agents which attempt to reach an equilibrium that is close to the optimum are more susceptible to delay-initiated oscillations. Fig. 9 shows the average mixed strategy (n=30000) for two cases of β in $\mathbf{D}_1$. For τ=0 (solid lines), the average strategies approach the optima for low α, that is, the players learn the optimal strategies for sufficiently small parameter values (as in [13]). At a fixed α, a smaller value of β degrades the performance - it is best to have a large distance between the parameter values.

Two examples of delay (dotted lines) are also shown for $\beta = 0.2$. As the delay increases, the performance decreases, however, there is a sufficient value of α for which the behavior approximates the instantaneous case. At the point where the delay and instantaneous cases agree, the persistent oscillations give way to damped oscillations, hence a stable, but suboptimal, equilibrium. For example, at β=0.2 and τ=500, values of $\alpha < 0.045$ initiate persistent oscillations whereas larger values have stable, but suboptimal, solutions. For τ=100, much smaller values of α can be applied and the result is a stable solution that is closer to the optimum.

As the delay goes to zero, the only unstable parameter setting is α=0 (the linear reward-inaction algorithm L_{R-I}, which has been shown to oscillate in the non-delayed version of (5) [13]).

5.2 Stability in Single-Level Games

The appendix shows linear stability analysis, with the results valid for any mixed-strategy equilibrium game. The result allows agents (or designers of agents) to know in advance the parameter settings that cause instability. Therefore, agents can achieve a solution that is as close to the optimum as possible without sacrificing stability. Alternatively, agents can decide how close to the optimum is desirable, then communicate sufficiently to achieve the tolerated amount of delay in information.

We now consider the results in the context of some specific games and parameter settings. Fig. 10 shows the instability delay, $\tau_2 = f(\alpha, \beta, \theta = 0.01, \mathbf{D}_1)$, that results from three different types of calculations. First, individual data points represent the onset of oscillations (the *observed* instability delay τ_o) as determined by (5), the strategies are examined over long runs at incremental delays to determine

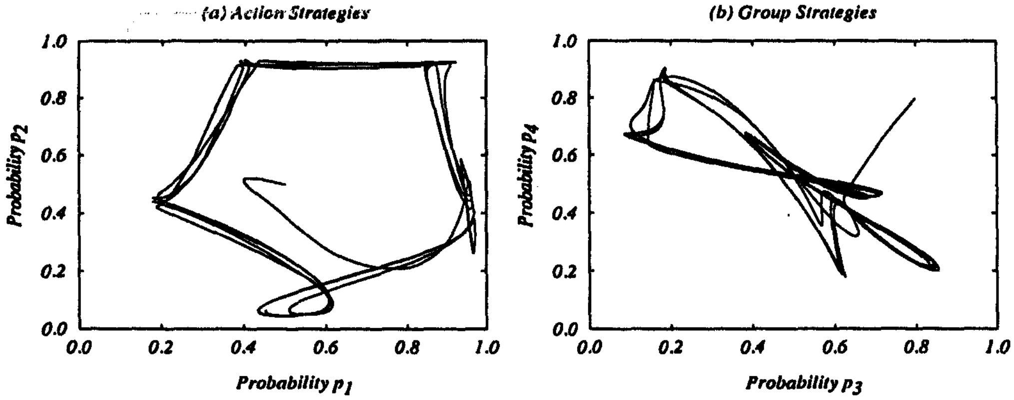


Figure 8: Chaotic Regime for Game 2 at High Delay ($\tau = 1000$)

which delay is sufficient. Second, the solid lines are the solution to the predicted behavior (18) assuming that the equilibrium is the game-theoretic solution $\mathbf{p}_{opt}$, a simple *a priori* calculation [13]. The predicted behavior fails (by 20-25%) as α increases, as to be expected from using the wrong equilibrium. However, applying the expected equilibrium $\mathbf{p}^*$ to (18) yields the results in the dotted lines (the *predicted* instability delay τ_p). This approximates within 5% the observed results. In general, the delay required to initiate persistent oscillations increases with α and decreases with β. This is exactly *opposite* the conditions that insure an equilibrium that is close to the value of the game. That is, parameters that are likely to lead to the optimal equilibrium in the non-delayed case are more likely to initiate instabilities in the delayed case. This is the trade-off suggested in Fig. 9 between optimality and stability - the choice of parameters is an implicit statement about which goal is more important. Unfortunately, there does not exist some minimum level of communication to guarantee stability. The minimum can only be decided in the context of the chosen parameters.

Fig. 10 shows that the data for β=0.1 and β=0.2 are approaching asymptotes (this is the case for the other values of β, but the asymptotes are not included in the domain of the graph). As α increases beyond this asymptote, the strategies are stable. This defines a parameter setting that guarantees the system will be stable, independent of the delay in the system, albeit at an equilibrium that is distant from the optimum.

The effect of θ is not shown but $\partial\tau$ is a measure of the relative delay in the system. In fact, the increase in the instability delay is proportionate to the decrease in θ, since the players do not make large steps in their decision-making process. However, a small θ also means that the players take longer to reach equilibrium.

5.3 Stability in Multi-Level Games

In this section, an approximation is used to determine the amount of delay τ_2 required to initiate persistent oscillations when the agents have a choice between games. The technique involves linearizing in the neighborhood of the equilibrium $\mathbf{p}^*$ (as in the appendix) and the assumption that p_3 and p_4 are constant and equal to the equilibrium values in $\mathbf{p}^*$, that is, $c = c^*$. This implies that we ignore the partial derivatives with respect to these variables. The analysis determines the stability of the average game induced by the equilibrium value for the group decision.

Table 1: Instability Delay: Predicted (τ_p) versus Observed (τ_o)

game	α	β	θ	c^*	τ_o	τ_p
2	0.02	0.80	0.1	0.2374	33	34
2	0.02	0.40	0.1	0.2417	145	148
3	0.01	0.10	1.0	0.6564	18	22
3	0.02	0.10	1.0	0.4812	52	51
3	0.01	0.05	1.0	0.4806	106	102
3	0.01	0.05	0.5	0.4793	218	203

The results are shown in Table 1 with the cases from Figs. 5 and 6 included. There is close agreement between the observed and predicted values of τ_2 and we can draw the same three conclusions as before: τ_2 increases with increasing α, decreasing β, and decreasing θ. The data suggests that ignoring the partial derivatives with respect to p_3 and p_4 did not hinder the analytic prediction even though these probabilities oscillated in game 3.

6 Large Systems

The primary emphasis has been placed on delays in 2-player multi-level games with learning automata strategies. The question arises: What happens in large systems? We are going to make some simplifications. In a sense, we are more interested in the behavior of large systems, independent of the search strategy employed to reach a consensus

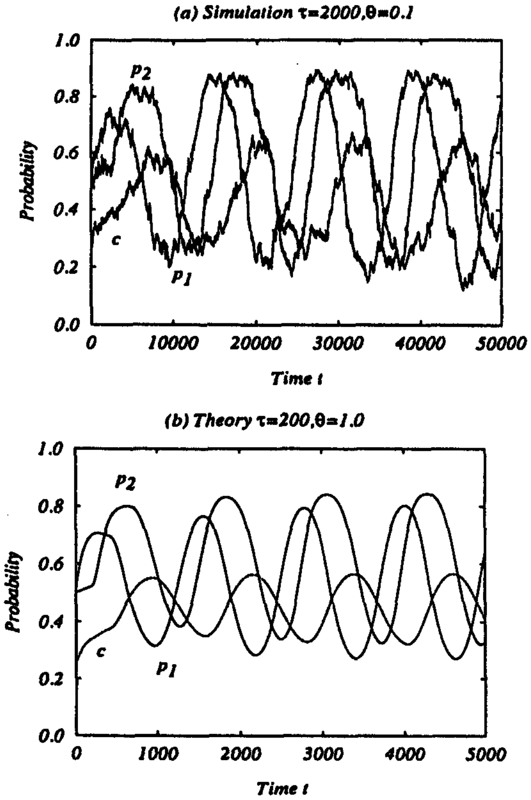


Figure 6: Oscillations in Action and Group Strategies of Game 3

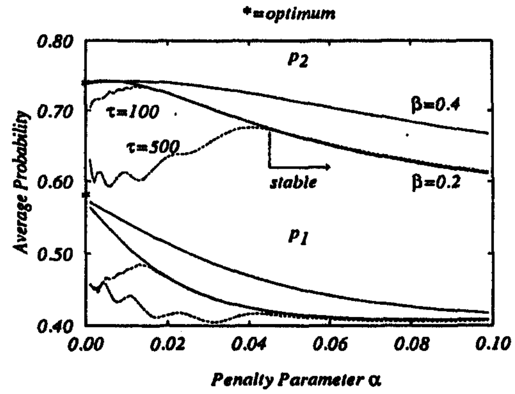


Figure 9: Effects of Learning and Delay on Average Strategy

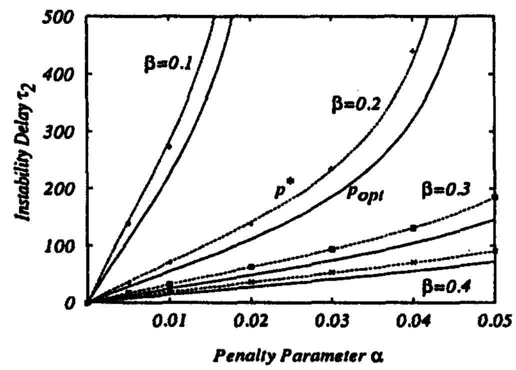


Figure 10: Stability Boundary

on group interactions. We are not concerned with the specific feedback rules or other decision rules available to the agents.

Consider n agents, with agent i willing to cooperate $c_i(t)$ of the time. This is the analog of the high-level probabilities used previously to decide group formation. Let us assume that an arbitrary decision process exists whose eventual goal or equilibrium concerning group interaction is c^*, the analog of the equilibrium value of the clustering parameter. That is, the agents should cooperate c^* fraction of the time. Let this value be known to all agents; the uncertainty lies in how the autonomous agents will achieve this unified goal based on individual decisions. We assume that a group is formed if, and only if, all agents are willing to join the group. That is, $c(t) = \prod c_i(t)$ is the likelihood of group formation and one way of achieving the goal c^* is by equal participation: $c_i^* = \sqrt[n]{c^*}$ (the result turns out the same if an additive rule, $c_i^* = c^*/n$, is used).

To simplify the analysis, we assume that a linear differential equation captures the search process and that agent i incrementally adjusts its own $c_i(t)$ toward the unified goal based on its view of $c_j(t-\tau)$ for all other agents j. This is not a specification of an algorithm. Indeed, the algorithm may be very complex, our assumption is that the resultant dynamics of the algorithm can be represented by this incremental search process. If agent i believes, based on aged information, that some of the agents are not participating enough, then it will tend to increase its own participation to help the overall interaction reach the goal c^* (and vice versa for a surplus). Agent i's individual goal is then:

$$g_i(t) = c^* \left[\prod_{j\neq i} c_j(t-\tau)\right]^{-1}, \qquad (7)$$

and the dynamics are:

$$\dot{c}_i(t) = \theta\left[g_i(t) - c_i(t)\right], \qquad (8)$$

where θ is a step size parameter which determines the rate at which c_i approaches agent i's participation goal g_i (from above or below). Note that the partial derivative of g_i with respect to any c_k, evaluated at $c_k^* = c_j^* = \sqrt[n]{c^*}$, is simply -1. The linearized equation for stability, with a rescaling of time, becomes

$$\dot{x}_i(t) = -x_i(t) - \sum_{j\neq i} x_j(t-\theta\tau), \qquad (9)$$

which has the exact form found in the analysis of delayed analog neural networks [15], [16] (and many other situations). The resultant stability boundary is:

$$\tau_2 = \frac{\cos^{-1}(1/(1-n))}{\theta\sqrt{n(n-2)}}, \tag{10}$$

where the inverse cosine takes its value from the interval $[\pi/2, \pi]$.

Naturally, in large systems, there tends to be more communication, say via messages. The complexity of the system grows by n^2, that is, there may need to be many more communication paths. The intelligent design of these paths, and the appropriate information to exchange [1], may help to reduce this growth. However, (10) shows that there is a requirement for even more communication related to the age of information and stability. For large n, we can see that τ_2 is inversely proportional to n. This means that the tolerance to instability decreases with n, which implies that the potential for communication overhead is of order n^3. Agents must communicate more frequently and/or networks must have less latency to reduce the average age of information. Of course, the question of a unique delay τ_{ij} between i and j arises in large systems. The current research in delay differential equations shows that this is a difficult case, even for systems with just two unique values of delay [20].

The stability boundary in (10) is inversely proportional to the search rate θ. This leads to the same, and not surprising, conclusion as in the specific example of learning automata: faster searching implies lower tolerance to delay-initiated instability.

7 Conclusions

A model has been presented with uncertainty in actions, group dynamics, payoffs, and state information. Learning automata achieve equilibrium in the particular cases examined with instantaneous information. This means that an agent successfully employs an automaton at each of the two levels. However, with delays in the system, the behaviors may exhibit damped or persistent oscillations and the onset of chaotic regimes. The analysis of the general case yields the delay required to initiate persistent oscillations; unfortunately, the parameter settings that decrease the likelihood of instabilities also increase the likelihood that a suboptimal equilibrium will result. As agents strive for optimality, it is more likely that instabilities will arise due to delays. This illustrates the fundamental problem of seeking the optimum strategy without being misled by delayed information.

Of course, the agents may have some control over the delays in the system by adjusting how often they communicate or even by restructuring the network. In this case, it is advantageous for the agents to reduce the age of information so that stable and optimal solutions are possible. The analysis is useful because it predicts the boundary between stability and instability. For a given set of learning parameters, a minimum level of communication can be established to insure stability. However, as the penalty parameter goes to zero, the tolerated delay also goes to zero. There is also a parameter setting that guarantees stability, regardless of the delay, but the solution will typically be far from optimal.

The problems are only exacerbated in large systems. We showed that a simple generalized search method creates an inverse relationship between size and the tolerated delay. This sensitivity to instability is similar to results in ecological models [21] and suggests that the results can be applied in various situations.

Appendix

The goal is to predict the onset of persistent oscillations using linear stability analysis [17] which, for the stable attractor in single-level games, is equivalent to determining Lynapunov exponents.

Let $k \in \{1, 2\}$ and

$$X_k = \theta\frac{\partial W_k(\mathbf{p}^*)}{\partial p_k}, \quad Y_k = \theta\frac{\partial W_k(\mathbf{p}^*)}{\partial p^\tau_{3-k}}.$$

Linearizing [4], [17] in the neighborhood of the equilibrium $\mathbf{p}^*$ determines the delay conditions to initiate persistent oscillations. The system of equations is

$$\frac{d\delta p_k}{dt} = X_k\delta p_k + Y_k\delta p^\tau_{3-k}, \tag{11}$$

and assuming solutions of the form $\delta p_k(t) = A_k e^{\lambda t}$,

$$\lambda A_k e^{\lambda t} = X_k A_k e^{\lambda t} + Y_k A_{3-k} e^{\lambda(t-\tau)}. \tag{12}$$

Eliminating common terms and rearranging,

$$\lambda - X_k = (A_{3-k}/A_k)Y_k e^{-\lambda\tau}. \tag{13}$$

This system yields:

$$(\lambda - X_1)(\lambda - X_2) = Y_1 Y_2 e^{-2\lambda\tau}. \tag{14}$$

Let $\lambda = r + iw$. There are an infinite number of discrete solutions and those parameter settings that yield only negative real parts are stable [4], [17] (with perhaps damped, but not persistent, oscillations). That is, marginal stability occurs at $r = 0$. The stability boundary can be determined by substituting $\lambda = iw$ in (14), applying Euler's formula, and solving for the real and imaginary parts:

$$\cos(2w\tau) = (X - w^2)/Y, \tag{15}$$

$$\sin(2w\tau) = (X_1 + X_2)w/Y, \tag{16}$$

respectively, where $X = X_1X_2$ and $Y = Y_1Y_2$. Dividing (16) by (15),

$$\tan(2w\tau) = \chi = \frac{(X_1 + X_2)w}{X - w^2} \tag{17}$$

and the instability delay (sufficient to initiate persistent oscillations) is:

$$\tau = \tau_2 = \tan^{-1}(\chi)/2w, \tag{18}$$

where the inverse tangent takes its value in the interval $[0, \pi/2]$.

Adding the squares of (15) and (16),

$$u^2 + Bu + C = 0, \quad (19)$$

where

$$u = w^2,\ B = X_1^2 + X_2^2,\ C = X^2 - Y^2,$$

hence $w = \pm\sqrt{u}$. The single solution to the quadratic equation is

$$u = \frac{-B + \sqrt{B^2 - 4C}}{2}, \quad (20)$$

as the other solution fails to insure a real (the only type of solution) for w (note that $B > 0$).

References

[1] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "The utility of communication in coordinating intelligent agents," *Proc. 9th Natl. Conf. on AI*, pp. 166–172, July 1991.

[2] K. Narendra and M. Thathachar, *Learning Automata: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] T. Hogg and B. A. Huberman, "Controlling chaos in distributed systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 1325–1332, Nov./Dec. 1991.

[4] J. O. Kephart, T. Hogg, and B. A. Huberman, "Dynamics of computational ecosystems," *Phys. Rev. A*, vol. 40, pp. 404–421, July 1989.

[5] R. Axelrod and W. Hamilton, "The evolution of cooperation," *Science*, vol. 211, pp. 1390–1396, Mar. 1981.

[6] E. Jensen, "Decentralized executive control of computers," in *Proc. IEEE Intl. Conf. Distr. Comput. Syst.*, pp. 31–35, 1982.

[7] J. Halpern and Y. Moses, "Knowledge and common knowledge in a distributed environment," *J. ACM*, vol. 37, pp. 549–587, July 1990.

[8] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "A decision-theoretic approach to coordinating multiagent interactions," *Proc. 12th Intl. Joint Conf. on AI*, pp. 166–172, 1991.

[9] A. Glockner and J. Pasquale, "Coadaptive behavior in a simple distributed job scheduling system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 902–907, May/June 1993.

[10] E. Billard and J. Pasquale, "Dynamic scope of control in decentralized job scheduling," in *Proc. IEEE Intl. Symp. Autonomous Decentralized Syst.*, pp. 183–189, 1993.

[11] E. Billard and J. Pasquale, "Effects of delayed communication in dynamic group formation," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1265–1275, Sept./Oct. 1993.

[12] E. Billard and J. Pasquale, "Adaptive coordination in distributed systems with delayed communication," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, Apr. 1995.

[13] S. Lakshmivarahan and K. Narendra, "Learning algorithms for two-person zero-sum stochastic games with incomplete information:A unified approach," *SIAM J. Control and Optimization*, vol. 20, pp. 541–552, July 1982.

[14] P. Gmytrasiewicz, E. Durfee, and D. Wehe, "Combining decision theory and hierarchical planning for a time-dependent robotic application," *Proc. 7th IEEE Conf. on AI Applications*, pp. 282–288, Feb. 1991.

[15] C. M. Marcus and R. M. Westervelt, "Stability of analog neural networks with delay," *Phys. Rev. A*, vol. 39, no. 1, pp. 347–359, 1989.

[16] J. Bélair, "Stability in delayed neural networks," in *Ordinary and Delay Differential Equations* (J. Wiener and J. Hale, eds.), New York: John Wiley & Sons, 1992.

[17] J. D. Farmer, "Chaotic attractors of an infinite-dimensional dynamic system," *Physica D*, vol. 4, pp. 366–393, 1982.

[18] R. Bellman and K. L. Cooke, *Differential-Difference Equations*. New York: Academic, 1963.

[19] J. Wiener and J. Hale, eds., *Ordinary and Delay Differential Equations*. New York: John Wiley & Sons, 1992.

[20] J. Hale and W. Huang, "Global geometry of the stable regions for two delay differential equations," *J. Math. Anal. Appl.*, vol. 178, pp. 344–362, 1993.

[21] R. May, "Will a large complex system be stable?," *Nature*, vol. 238, 1972.