

16 May 88

*

A COMPOSITE INFORMATION SYSTEM
FOR
THE SLOAN PLACEMENT OFFICE

by

Laurence Stanley Kooper

B.A., Computer Science
New York University
(1986)

Submitted to the Sloan School of Management
in Partial Fulfillment of
the Requirements of the Degree of
Master of Science in Management

at the

Massachusetts Institute of Technology

May 1988

© Laurence S. Kooper 1988
All Rights Reserved

The author hereby grants to M.I.T. permission to reproduce and to
distribute copies of this thesis document in whole or in part.

Signature of Author
Alfred P. Sloan School of Management
May 13, 1988

Certified by
Y. Richard Wang
Assistant Professor, Management Science
Thesis Supervisor

Accepted by
Jeffrey A. Barks
Associate Dean, Master's and Bachelor's Programs

A COMPOSITE INFORMATION SYSTEM
FOR
THE SLOAN PLACEMENT OFFICE

by

Laurence Stanley Kooper

Submitted to the Alfred P. Sloan School of Management
on May 13, 1988 in partial fulfillment
of the requirements for the Degree of
Master of Science in Management

ABSTRACT

Information systems which require the integration of separate databases, referred to as composite information systems (CIS), are becoming increasingly important as businesses wish to protect their large investments in hardware, software, and training.

The developers of a CIS must give due attention to strategic, organizational, and technical factors to achieve the system's goals.

This study describes a prototype CIS written for the Sloan School of Management Placement Office. It finds that using relational database software, a small team can implement such a system in a short time. Management of the development effort was found to benefit from voluntarism but suffered from lack of effective monitoring. It is concluded that the developers overcame organizational obstacles through informal communications, flexibility, and familiarity with the application to be automated.

Thesis Supervisor: Dr. Y. Richard Wang
Title: Assistant Professor of Management Science

ACKNOWLEDGEMENTS

I would like to express my thanks for all of the assistance I have received on this project.

Richard Wang and Stuart Madnick were excellent advisors who gave generously of their time. They inspired me to do my best.

Linda Stantial of the Sloan Placement Office supported the project and provided time and valuable information.

David Rogers contributed his experience, valuable insights, and help at several critical moments.

Dave Horton's related work and level head were an inspiration.

Alec Champlin and T.K. Wong were always willing to take time to bring me up to date on their latest work.

Drew Banks, Andrea Flamburis, and Lisa Tener contributed valuable ideas and support, and devoted a great deal of time to making PAS real. They worked beyond the call of duty. I thank them for their enthusiasm and patience.

CONTENTS

ABSTRACT	2
ACKNOWLEDGEMENTS	3
1 INTRODUCTION	6
1.1 Sloan's New Systems Environment	6
1.2 Placement Office Strategy	6
2 PERSPECTIVES ON THE ORGANIZATIONAL ISSUES	11
3 THE PAS SYSTEM	17
3.1 Core System Functions	17
3.2 Designing the Prototype: The Management Process	18
3.3 Implementation and Design Decisions	20
4 LOGICAL CONNECTIVITY: PROBLEMS AND SOLUTIONS	28
4.1 Introduction	28
4.2 Queries	29
4.3 Other Logical Connectivity Problems	33
5 CONCLUSIONS AND FUTURE WORK	35
5.1 Conclusions	35
5.2 Future Functionality	35

APPENDIX A: PAS: A Sample Session	38
APPENDIX B: PAS System Specifications	44
B.1 Tables in Recruit DB	44
B.2 Overall Logic	48
APPENDIX C: Daily PAS Updates	50
BIBLIOGRAPHY	56

Chapter One

Introduction

1.1 Sloan's New Systems Environment

In December, 1987 the Sloan MIS faculty released a Vision Statement on the future of Sloan Information Systems and Services (SISS) [MAD87]. The plan envisions a network which will intelligently connect all of Sloan's computers and databases, as well as providing the ability to access external data.

Placement activities should be and are an important part of this new environment. For most of Sloan's students, improved job prospects upon graduation are one of their main reasons for enrolling. Students expend a great deal of time and energy in the job search and expect high quality assistance, support and information from the school. In turn, Sloan has high expectations of its students' success in the marketplace, which helps to reflect prestige on Sloan and MIT and makes possible future alumni support and the hiring of future graduates.

Among the goals expressed in the Vision Statement are "improved productivity through the use of information systems," "training in online databases," and "maximizing the usefulness. . . of information."

1.2 Placement Office Strategy

A group led by Andrew Trice produced a requirements definition for a Placement Office system [TRI87]. According to this document, the mission of the Sloan Placement Office (SPO) is:

to facilitate and optimize the satisfaction of Sloan students and graduates with their careers, and the satisfaction of employers with their efforts to recruit and hire Sloan graduates.

Placement Office actions and statements reveal that their mission goes well beyond the job search process. The more global aspects of the Placement Office's mission are:

To explore the relation between academic studies, students' interests and the employment market, and

To enhance the prestige of Sloan through successful placement of graduates, thus attracting higher quality students to the School.

Indeed, placement is an integral part of the "M.I.T. School of Management" concept statement [MIT88]. The statement, part of Dean Lester Thurow's strategic marketing plan to compete with Harvard and Stanford business schools in attracting the best candidates, sets a goal of 75% of students to pursue careers in multinational companies and 65% in technology-based companies. This points up the crucial and perhaps overlooked fact that for Sloan, placement is not only a private transaction between students and the marketplace but also a culmination of the school's educational purpose -- a "use" of the "outputs."

If Sloan is truly to reach its goal of creating managers who can understand technology, globalization, and organizational change, the Placement function will be the "front line" on which its teachings are tested.

This thesis describes the Placement Assistant System (PAS). The strategic purpose of PAS is to further all of these goals: to improve productivity, maximize the usefulness of information, and to aid the placement office in putting the "outputs" of Sloan to good use.

The overall objectives of the Sloan Placement Office (SPO) are:

- To provide career counseling

- To develop a receptive employer base
- To coordinate recruiting and interviewing
- To maintain search information and resources
- To report placement results

Sloan Placement Director Linda Stantial often reminds Sloan students that on-campus recruiting is only one part of a student's job search. Only 45-60% of second-year and 15-20% of first-year students find positions through on-campus recruiting. (See [MAS87a], [MAS87b]). By the same token, work related to on-campus recruiting forms only one part of the Placement Office's activities. Preparing workshops and presentations, counseling individual students, and acting as liaison with recruiters and the media are important and time-consuming parts of their work.

However, on-campus recruiting is the most data-intensive activity which the Placement Office performs. As such, it is the area in which an information system can be most cost-effective. While students are conscious that they bear ultimate responsibility for finding a suitable job, on-campus recruiting forms the largest part of what they expect from the Placement Office. In addition, on-campus recruiting is the area in which students' needs are the most "standardized." If information flow related to on-campus recruiting can be made more efficient both from Placement Office to students and from students to Placement Office, the latter will have more resources freed to provide "non-standard" assistance to recruiters, student groups, and individual students.

Currently, the SPO maintains its information in paper databases. It also keeps and maintains data on a Wang word processing system. Though well suited to producing letters and documents, this is an extremely cumbersome and inflexible system to use for maintaining lists and tables of constantly changing, high-volume data that must be sorted and related in various ways. Their current

work setup forces the SPO to do the same work multiple times. For example, when a new firm comes to recruit, its name must be manually placed on many different lists (e.g. sorted by location, sorted by job function, etc.).

Although students have been successful in finding jobs, due to the scattered nature of the data and the difficulty of making timely updates the process of getting information and learning procedures is difficult and time-consuming.

Recent staff turnover has forced the SPO to face the need to find and train new staff of its own.

The solution which PAS proposes to these problems is to give the SPO staff more time to develop an employer network and provide students with individual counseling. The system aims to reduce the staff's paperwork burden and to facilitate students' job search. The job search is above all a human activity, and involves face-to-face human functions that no computer can perform. PAS is one of a series of steps, the first small step for the SPO.

Benjamin, DeLong, and Scott Morton [BEN88] discuss the the "paradox" in the design and implementation of inter-organizational systems (IOS). "In order to gain organizational acceptance of the system," they found, "developers. . . intentionally minimized any changes in the work processes affected by these systems." Thus, the systems are accepted but benefits are not as great as they could have been had work processes been redesigned. PAS's developers also faced this paradox and are resolving it in a similar way. The first step provided by PAS will be to save labor. Much as the replacement of typewriters with word processing and of calculators with spreadsheets provided strategic advantage by changing work styles, the productivity gains produced by PAS should lead to the goal of improved recruiting results.

There are two types of IOS: transaction processing systems and task support systems [BEN88]. PAS contains elements of both of these types of system.

The transaction processing component will aid the SPO by automating the production of routine documents such as:

- Company Presentation Schedules
- Recruiting Schedules
- Daily Interview Schedules
- Open interview winners
- The Placement Report (Summarizing job offers, salaries, etc.)

The task support component is intended to help students help themselves, by providing flexible, up-to-date, on-line access to information. Recruiting information will be accessible via campus PC's 24 hours a day. Alumni information will be available. In conjunction with recruiting information, it will allow students to identify Sloan alumni who work at companies they are interested in, whether or not the company is coming to recruit on campus. In addition, the system will store students' previous job experience, thus enabling students to share information with peers about the employment world.

Chapter 2 summarizes the organizational issues. Chapter 3 describes the system functions and details how the design effort proceeded. Chapter 4 discusses some problems in logical connectivity and their solutions. Concluding remarks are made in Chapter 5.

Chapter Two

Perspectives on the Organizational Issues

Information systems which require the integration of separate databases are referred to as *composite information systems* (CIS) [MAD88b]. In implementing any composite information system, it is necessary to identify potential organizational problems and propose solutions (or evolve solutions through trial and error). (See [OSB87], [MAD88b].) The PAS application is a particularly interesting example, with many groups having an interest in the system and being crucial to its success.

Students are important to the success of the system in that as one of the ultimate user groups, the system's planning and design should reflect their needs and suggestions. However, students have little effect on SPO policy as long as they are receiving adequate placement services. Once accepted and enrolled at Sloan, the Placement Office is their "gateway" to employers. Thus students are a captive audience who are not free to go elsewhere for comparable services. If the system as implemented ill serves students' needs, they will still have no choice but to use it.

However, there are other stakeholders who have more power to influence the system development and outcome. They are:

- The Sloan Placement Office (SPO)
- The Sloan Master's Office (SMO)- also has great need for IS work
- The MIT Alumni Office - Keeps alumni information and is sensitive about "mailing lists" (i.e. access to alumni funds)
- Central Administration (i.e. the Bursar's office, Registrar's, etc.) - Keep data about students including Sloan's. Some data may be redundant with

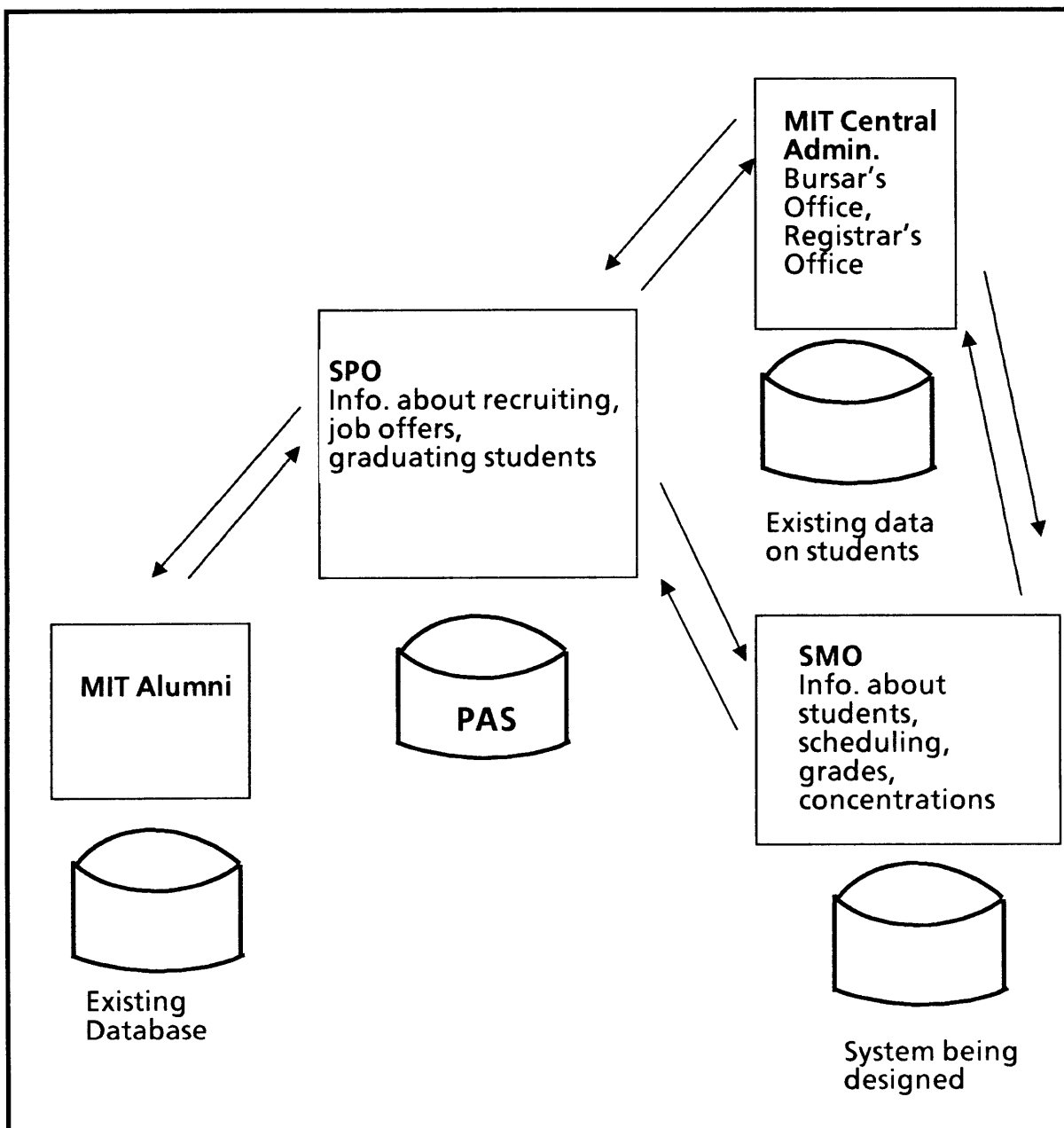


Figure 2.1: Organizational Units with a Potential Stake in PAS

contemplated databases of SMO and SPO. However, combining potentially raises issues such as security, data integrity, and turf.

- The Sloan MIS faculty - Emphasize research and pushing frontier of knowledge. View school as a laboratory for testing new ideas.

Figure 2.1 depicts these groups and their databases, excluding the faculty since their database is represented by that of the SMO.

The paradigm for this situation is not “corporate strategy” but rather “loosely coupled organization” (cf. [WEI76]). A loosely coupled system is one in which the elements have a certain degree of freedom and autonomy and that is beneficial to the organization. In the “corporate strategy” paradigm, a powerful leader (the CEO, CIO, a planning department, a vice president) gives the orders and provides the overall leadership. In a loosely coupled organization, groups and offices are more free to pursue their own aims. High authorities lead by persuasion, suggestion and example. Culture and atmosphere provide as much direction as strategy does.

The PAS development team had some stake in all of these areas. We were students with some need for placement assistance, we dealt with the Sloan and MIT administrations, and we were interested in MIS in theory and practice. As future alumni, we had a stake in the Alumni Office. This enabled us to serve as a disinterested buffer between the various loosely coupled factions.

The development team faced the paradox described earlier by Benjamin, DeLong, and Scott Morton [BEN88]. In order to gain acceptance for the system, we promised that the SPO’s operations and relationship with students would not be changed. However, down the road at some point such changes may prove necessary to get full benefit from the system.

Our first contact with the SPO on this project was inauspicious. We planned to distribute a survey to Sloan Master’s students to determine their needs and requirements for placement information. The SPO denied permission to distribute this survey. Their argument was that job placement is an important, emotional issue to students. The SPO viewed students as their clients, and wanted no interference in that client relationship. The SPO did not

want students' expectations to be raised unduly. They also wished to avoid raising student fears about interruptions of placement services or problems with data integrity, problems that often occur as an office converts from manual to computer processing.

Though we were not yet aware of this, there had been prior attempts to automate the functions of the SPO. Previous projects had been shelved without true implementation, leaving the SPO disappointed and suspicious of wasting time in blind alleys.

At this point the team thought the project was at an impasse. Though most of the team were discouraged, one member, Andrea Flamburis, thought that the main problem was *communication*. Andrea felt it was better to approach Linda Stantial, the SPO director, directly rather than attempting to circumvent her. Andrea realized that it was crucial to view Ms. Stantial as *our* client. Further, Andrea realized that an informal, "off-line" approach would work best at defusing resistance. At a weekly beer party, Andrea approached Ms. Stantial and suggested that she meet directly with the group and discuss her concerns. Ms. Stantial agreed.

At the meeting, she told us that she welcomed the idea of automating the SPO's operations. However, she had two major concerns: (1) lack of time (due to reduced staff) to assist our project, and (2) student expectations and anxieties, as described above. Because Ms. Stantial looked favorably on our overall goals, we made commitments that resolved these concerns. First, we agreed to develop the system largely on our own. The only time we asked of SPO personnel was time to demonstrate our prototype once written and to critique our documentation. Second, we agreed not to raise student expectations. We would emphasize to students that the project was only a prototype, and SPO operations would not change. Thus, we resolved Benjamin, DeLong, and Scott Morton's IOS paradox,

while deferring to the future the painful questions of organizational change (such as change in SPO work styles) implied by a true implementation.

Ms. Stantial stated that the SPO activity she would most welcome having automated was the Priority Card system, by which second-year students bid for slots on "open" interview schedules. She said that the mechanical operations of verifying student qualifications (such as citizenship and degrees), sorting cards, and reporting interview winners "eat up a lot of time." We therefore decided to make the Priority Card system one of the main goals of our prototyping effort.

After the meeting with Ms. Stantial, we proceeded with system development independently. Towards the end of the project, we demonstrated the system for Ms. Stantial. She was favorably impressed, and expressed the desire for a phased implementation of the system, starting with the Priority Card system.

[TEN88] contains a further discussion of the organizational problems of PAS and their solutions. The solutions that we discovered to be effective were:

- 1) A direct, informal approach
- 2) Familiarity with the application (since we were SPO clients)
- 3) Willingness to adjust goals (from quick implementation to prototype/study).

Osborn [OSB87] has identified three areas in which to seek success in composite information systems. The areas and application to this case are:

Bidirectional Benefits: Discussed above. For SPO - more productivity, being able to provide better service to recruiters and students, fulfill Sloan strategic goals. For faculty - opportunity to apply research to a real setting.

Cooperative Payoff: Improved coordination among MIT offices, less duplicated work.

Asymmetrical Control: Definitely an issue here, since SPO has a lot to gain from a successful system but has little time or specialized knowledge to maintain

it. One concern of the SPO is that no third party has yet appeared with the necessary long-term commitment to making the system work and maintaining it. Students are transient, faculty are otherwise engaged and the administration has yet to make the system a priority.

In order for PAS to become a truly successful system, the ongoing development effort must face these issues.

Chapter Three

The PAS System

3.1 Core System Functions

Reflecting the goals of PAS as discussed in Chapter 1, the core functions that PAS performs are embodied in the Recruit Database (RecruitDB).

From the student's point of view, this embraces the functions of:

- Choosing firms to submit resumes to
- Choosing firms to submit cover letters to
- Choosing firms to submit priority cards to
- Submitting priority cards
- Signing up for an interview time slot
- Searching for job opportunities using criteria such as location, position, firm size, etc.
- Each student will be able to produce an individualized schedule of interviews including date, time, firm, and position.

The following Placement Office functions are supported:

- Input and maintenance of recruiting dates, contact information, etc.
- Room scheduling
- Verifying priority cards and determining open interview "winners" (See Appendix B for a detailed description of the operation of the Priority Card system.)
- Production of reports/documents such as:
 - Recruiting Schedules (by company, by date)
 - Employers Recruiting at Sloan (by industry, function, and company)
 - Company Presentation Schedules
 - Daily Interview Schedules

Placement Report

Appendix A contains a sample PAS session, and Appendix B system specifications.

3.2 Designing the Prototype: The Management Process

3.2.1 What went right?

(A) Voluntarism

Rather than assigning specific tasks to everyone, I called upon them to volunteer. I wanted to make the commitment process informal, similar to the “signing up” described by Kidder at Data General [KID81]. This method was effective since these were highly motivated, energetic people. Overall, I gave the group free rein to accomplish their tasks. We held meetings about once a week, plus we would often speak informally. We also communicated through written memos. In fact, when I realized how many memos I was writing to the team I began to structure them as a “newsletter” called *Daily PAS Update*. (See Appendix C for some examples.) Keeping them informed on my progress involved not just giving them copies of what I had written, but sharing with them my thought processes and asking them for help with problems I was in the process of resolving. I also used the memos to report minutes of meetings and put in writing the different things everyone was working on. The other team members also gave feedback to me. Since they were not spending as much time on the project as I was, they helped to provide an “outsider’s view” which I lacked. In all, everyone involved in the project had a high level of commitment. I wanted to encourage the group’s creativity, since they truly were interested in helping the Placement Office solve its problems.

(B) Interim Public Demonstration

At the midpoint of the project, we demonstrated our system prototype for our MIS class. This was helpful both before (as an interim goal to aim for) and after (to help us improve the system).

3.2.2 What could have been improved?

(A) Human Factors

It is important to plan for the actual operation of the system. What will it look and feel like to the new user? For testing purposes, the person who wrote the user interface should not be the one who demonstrates the system. The tester should be someone who knows nothing about the system, preferably someone from the intended user group. The user should not need to sit in front of the machine with a user manual in order to understand how to use it.

(B) Timing

It is important to allow enough time for the system to be written. I didn't take into account the fact that my time horizon and commitment to the project were different from the other team members. I was devoting 75-90% of my time to this project. The other team members, however, had many commitments besides this one. Their other commitments involved a lot of small, short-term assignments. Thus, they were used to thinking in a "what's due tomorrow?" last-minute time horizon. The best way for me to circumvent this problem was to set interim deadlines. I did try to do this (see, for example, *Daily PAS Update* for April 5 in Appendix C), but if there was no public commitment such as a demonstration for outsiders I did not have the authority to enforce a deadline. This is one example of where voluntarism was not wholly effective.

I made the mistake of not allowing enough time to make mistakes and correct them. I should have adopted the philosophy expressed by Brooks [BRO75]

of “plan to throw one away.” Rather than taking the attitude of “get the product out,” I should have tried to learn as much as possible from the development process, and go through an improvements cycle.

(C) Communication and Monitoring

It is important to have written documents on what the user interface is expected to look and feel like. At meetings, I did discuss with the group what I expected and some of the things I wanted from the user interface, suggesting basic screen layouts and operation. However, I failed to put the specifics into writing. I delegated the task of writing the user interface. Though it was appropriate to delegate this task, it was a mistake to leave it to one person with no interference or criticism from other group members. People are often blind to the flaws in their own work. Even if the work is done by only one person, another should be there to provide support / criticism / a different view. I should have taken the role of chief architect (user’s agent) as described in [BRO75] and carefully specified external implementation.

3.3 Implementation and Design Decisions

3.3.1 General Principles

Our design team designed the system based on:

- discussions with SPO personnel including Placement Director Linda Stantial
- informal discussions with Sloan students, and
- our own experience as Placement Office clients.

Our guiding principles were:

- The system had to be easy to use. One of the main problems which students have in their job search is learning the Placement Office’s procedures, the location of various data, how to search for information and how to submit required information. Thus, we agreed that a well-written, attractive, user-

friendly interface was crucial to the system. In addition, even at Sloan some people are put off by computers and we didn't want students to be intimidated in an activity as crucial as the job search. Sloan students are impatient, busy people and need fast and easy access to information. A good user interface would be essential in winning over both the students and the SPO staff.

Achieving this goal was one of the more difficult aspects of developing the system. We initially implemented our prototype on the AT&T 3B2 computer using RDS's Informix SQL relational database management system (DBMS). Initial high hopes led to frustration as the limited flexibility of this software became increasingly clear.

- The system must reduce the Placement Office's workload. Although an initial training period would be necessary, the overall goal was for the system to make less work for them, not more.

- The system should allow its users (students, Placement Office personnel, and in future recruiters) to do things that they were unable to easily do before on their own.

- There needed to be clear responsibilities and commitment for maintenance, training, data entry, backups, and other utility functions.

- We wanted to avoid a repeat of the previous unsuccessful projects mentioned in Chapter 2. In order to do this, we followed the traditional MIT development philosophy of incrementalism and learning from use. We rapidly created and demonstrated a system which embodied the core functions. The system is intended to be a "building block." Users and design team will learn from experience and revise and add to the system accordingly.

- Enhancements and changes must be relatively easy to make. Rather than attempt to design a perfect system which then would be written, we followed the approach of rapid prototyping. We proceeded with all phases of system design

simultaneously -- requirements definition, specification writing, implementation, and testing. We chose this method of development to take advantage of *public testing* (cf. [SCH83]) and get early feedback from our constituencies. Since this was our philosophy and design method, we needed to ensure that the system was modular, well documented, and adhered to standard features of our DBMS (Informix-SQL) and operating system (UNIX).

- Data should be up-to-date and accurate.

- When the system “goes live,” student access to computers must be assured. We designed the system so that its entry point was the IBM or compatible PC. There are currently many PC’s available for student use on campus. In addition, many students own PC’s and could in future access the system by dialing in. In placement, time is critical since coordination with recruiting companies is needed. It is expected that there will be heavy demand for the system just before Placement Office deadlines, possibly raising problems of supply of PC’s and concurrency problems (multiple users trying to update the same data files simultaneously).

- Response time would have to appear reasonable to users.

- Down time for the system would have to be low.

3.3.2 Software Alternatives and Justification

General comment: There exists a tradeoff between flexibility / user friendliness and ease of coding / ease of change.

A procedural language such as Cobol, Pascal or C

Pros:

- Most flexible

Cons:

- Longest time to develop

- Need specific specifications before beginning
- Need programmers with specialized knowledge
- Harder to document
- Harder to change

Why not chosen:

- Time horizon of project.

Ada

Pros:

- Also flexible
- Makes it easier to separate project into parts, just specifying interfaces
- Has facilities similar to objects (packages)

Cons:

- Same as for other procedural languages

Why not chosen:

- Time, lack of specialized knowledge on team.

Unix shell, with SQL

Pros:

- Portability to any Unix machine
- Can make use of all Unix utilities and features, full access to machine's file system
- Can call C programs if need to (However, this leads to problems of who will maintain the C code)
- Can more easily call specialized utilities, such as CIS/TK (see Chapter 4 and [MAD88d])
- No compilation required

Cons:

- Difficult to program

- Programs are not as self-documenting as Cobol, Pascal, or Ada
- Report formats are not flexible, they are limited to SQL query output
- Difficult to make a user-friendly interface with input validation, exception handling, cursor control, help, messages, etc.
- Cumbersome and slow; some tasks require multiple file passes

Why not chosen:

- We felt it was not user-friendly enough.

Informix Formbuild (screen generator), **User-menu** (menu generator), **Ace** (report writer) (What was actually used)

Pros:

- Can quickly develop an application
- Menus are attractive looking
- Ace reports allow customizing of SQL output
- Underlying SQL is portable, though front end is not

Cons:

- Lack of flexibility. For example, one can't pass parameters between menus (such as a student's social security number). Want to ask student's name at beginning of session and not require student to re-enter it. Can't break out of menu logic, for example to handle exceptions. Can't route the output of a select into a customized screen.
- Formbuild - you are stuck with their interface (add, update, query, etc.) - unattractive looking, and difficult to understand for inexperienced users.
- Poorly written manuals - for example, no complete statement of syntax for an Ace statement - I had to write my own by looking through manual (see figure 3.1).
- Bugs - for example, manual says can call a submenu from Unix... you can't.

Why chosen:

- Time - could quickly develop the application
- Attractive menus
- We didn't realize how inflexible it was until we were well along in programming
- Unlike Unix shell, it naturally works together with SQL

statement:	simple-statement compound-statement
compound-statement:	FOR IF THEN ELSE WHILE
simple-statement:	LET NEED PAUSE PRINT PRINT FILE SKIP SKIP TO TOP OF PAGE

Figure 3.1: Syntax for an Ace statement

Informix 4GL (SQL application language)

Pros:

- More flexibility than Informix User-menu
- Freer logic in programs than User-menu
- Easy to make ring menus (Similar to Lotus 1-2-3)
- Developer can write two levels of help for user - one liners and full screens

Cons:

- Still a "sealed" system- can't make use of all Unix features
- Difficult to program
- Incompatible with Informix User-menu and Ace - Whole system would need to be rewritten

Why not chosen:

- Lack of time, software not available until last moment

- Incompatibility with User-menu made incremental change impossible - would have had to wholly rewrite system

Oracle (as implemented on IBM RT PC)

Pros:

- Same as for any other SQL implementation

Cons:

- No menu-building environment

Why not chosen:

- Not user-friendly

Easy SQL (as implemented on IBM RT PC)

Pros:

- More user-friendly than bare SQL

Cons:

- Inflexible. As with Formbuild, one is stuck with their standard interface

Why not chosen:

- We experimented with it and were dissatisfied with the standard interface.

3.3.3 Design Successes

1. Using a relational database was the right choice, since logic for reporting, maintaining, updating, sorting, etc. is all prewritten, yet DBMS is still flexible enough. The problem is finding a good "front end."

2. Deciding to build the priority card system was a design success since it won SPO support.

3. Basic table design proved to be good. Possible future problem with multiple divisions of same firm, e.g. Goldman Sachs. The SPO views separate

divisions as separate firms. If this is carried into the database it may lead to problems of connectivity with other databases such as students, alumni, Reuters.

4. Ability to do fast prototype helped us focus on desired system functions, rather than minutiae such as report formats. "Top-down" quality of starting with menu system let us quickly envision coherent set of system functions, which was clear to Ms. Stantial.

3.3.4 Management of Design

We should have explicitly stated our criteria for choosing platform and software. This would have helped us justify our decision and make a better decision.

Chapter Four

Logical Connectivity: Problems and Solutions

4.1 Introduction

Madnick and Wang [MAD88a], [MAD88d] discuss the problems involved in producing meaningful composite information from multiple disparate databases. In order to produce this information, it is necessary to solve problems in *logical connectivity*. The authors define logical connectivity as follows:

The process of resolving the semantic contradiction, inconsistency, and ambiguity that results from different assumptions made in disparate databases is referred to as logical connectivity. [MAD88a]

Logical connectivity is the process of answering a question in the spirit in which it was asked. To be truly logically connective, the answer must reflect the semantics (meaning of terms, units of measure, etc.) in the mind of the questioner.

Sloan students devote many hours of their busy schedules to the job search and interview process. A large portion of this time is spent searching for and processing information. As stated in Chapter 1, the major goals of PAS are to improve productivity, save time, maximize the usefulness of information, and maximize the satisfaction of students with their careers. Therefore, the designers and implementors of PAS should provide users (both students and Placement counselors) with logically connected answers to queries, in order to minimize the human processing time needed to make sense of information located in the different databases.

In [MAD88d], Madnick and Wang discuss three methods for resolving logical connectivity problems. They are:

- (1) Table lookup (also referred to as inter-database table (IDT) or mapping). This method is used to tie together identical or similar things with different names.
- (2) Functional mapping. These are procedures used to convert units of measure or retrieve information external to the database.
- (3) Heuristic reasoning. This is used when there is no clear-cut answer, but a set of rules may be used to come up with a good guess.

A system called CIS/TK, which will aid in implementing these solutions, is currently under development. See [MAD88d].

This chapter presents some examples of student queries, some of the logical connectivity problems they raise, and some of the ways in which these methods may be used to resolve these problems.

4.2 Queries

Query 1: Based on my area(s) of concentration at Sloan, what positions are likeliest to be desirable to me?

In order to answer this query, it is necessary to resolve inconsistencies between Sloan program concentrations and job functions. The inconsistency can be described as a *weak equivalence* (cf. [ELM87]), i.e. an approximate matchup. Figure 4.1 shows a suggested mapping to resolve this problem.

Query 2: What are the names of the alumni who have the same positions (perform the same job functions) as those companies are recruiting for on campus?

The problem here is that the Alumni database and the Recruit database have different levels of *granularity* (cf. [MAD88c]). Figure 4.2 shows the Alumni database's position table. It reports alumni positions along broad categories, such as Manager and Analyst. Recruiting positions, by contrast, are much more specific. Again, table lookup would be a good method for resolving this problem.

Sloan Program Concentrations	RecruitDB Job Function Codes
01. Accounting and Control	01. Accounting & Control
	02. Administration
02. Applied Economics	03. Applied Economics
03. Corporate Strategy, Policy, and Planning	04. Consulting
03. Corporate Strategy, Policy, and Planning	05. Corporate Strategy / Planning
04. Finance	06. Finance
	07. General Management
05. Health Care Management	08. Health Care Management
06. Industrial Relations, 13. Organization Studies	09. Human Resources
07. International Management	11. International Management
08. Management Information Systems	10. Information Systems
	12. Law
	13. Management Development Program
09. Management of technological innovation	19. Technology Management
10. Marketing	14. Marketing / Sales
11. Operations Management	15. Operations / Production
12. Operations research/ statistics	16. Operations research/ statistics
	17. Real Estate
14. System dynamics	18. System Dynamics

Figure 4.1: Mapping between concentrations and job functions

A table could relate, for example, Systems Analyst and Financial Analyst to Analyst.

Alumni DB Positions

Position Code	Position Name
00300	BUSINESS
00301	Chairman
00302	Chief Exec Officer
00303	President
00304	Vice President
00305	Dir/Head of Dept
00306	General Manager
00307	Manager
00309	Comptroller
00310	Treasurer
00311	Accountant
00313	Actuary
00314	Analyst
00315	Secretary
00316	Div Head/Supervisor
00317	Group Leader/Head

Recruiting Positions

Marketing Manager
 Systems Analyst
 Product Manager
 Materials Planner
 Financial Analyst

...

Figure 4.2: Alumni DB Positions
 Versus Recruiting Positions

Query 3: I want to find a job in New York City or its suburbs. What jobs are available?

Problem: The New York City metropolitan area contains parts of New York, New Jersey, and Connecticut but not the entire states. (Figure 4.3 gives some representative data). A solution could be to use zip codes to look up which cities are actually in the New York metropolitan area.

Positiontb

City	State	Zip
Hartford	CT	06152
New York	NY	10043
Bridgeport	CT	06601
Greenwich	CT	06830
Mt. Olive	NJ	07828
Norwalk	CT	06855
Newark	NJ	07102

Figure 4.3: Excerpt from Position table showing problem of Query 3

Query 4: What positions pay over \$50,000 per year?

The problem in answering this query is that semantics of salaries reported by recruiting companies vary greatly. These are all examples drawn from actual SPO job descriptions:

Base Salary

Competitive
 45,000-60,000
 45,000
 Depends upon individual
 (field blank)
 32,000 (minimum)
 Fully competitive

For entries such as “competitive,” “fully competitive,” and “depends upon individual,” it would be appropriate to use heuristic reasoning. Rules could be developed to approximate salaries using knowledge about industries, job locations, the student’s background (degrees, years of experience, and the like), and macroeconomic conditions. [HOR88b] contains a good approach to this problem.

Query 5: Do I have the necessary background for this job?

Problem: Language used in job descriptions does not always match up with that on students' resumés. To illustrate, here are two examples of job descriptions:

(1) Requirements: In-depth knowledge of Marketing discipline. Experience in product management, product development, strategic planning, advertising or sales management. Strong financial/ analytical skills, as well as strong business knowledge.

(2) Qualifications: A personal interest and experience in financial analysis or lending. An interest, personal skill, and preferably, experience in negotiating and dealing with customers on a face-to-face basis.

Again, the best method to solve this problem would be heuristic reasoning.

Examples of rules that could be used in the solution are:

If student has prior job in accounting, then student has experience in financial analysis.

If student has sales experience, then student has experience dealing with customers.

If student has technical undergraduate degree, then student has strong analytical skills.

An extensive rule set with rules such as the above could go a long way towards automating the matchup of students with suggested jobs, and save a great deal of human search time.

4.3 Other Logical Connectivity Problems

The design of PAS raised two other logical connectivity problems, which arose out of the SPO's procedures under their manual system. The SPO collects job function and industry data from recruiting companies, and it also collects this

information from graduating students about the jobs they have been offered and accepted. In both cases, different sets of codes are used.

Function Codes: There is one set (1-19) on the job description forms, and another set (1-12) on the offer/acceptance reports. (See Figure 4.1 for the 1-19 set.) All twelve of the (1-12) set are present in the (1-19) set. However, the numbers do not match.

Industry Codes: On job description forms, a 1-45 coding scheme is used. Offer/acceptance reports use codes from 1-31. Though some correspond, some are combined in different ways and differently stated. Again, numbers do not match. In addition, there is the question of relating these industry codes to Standard Industrial Classification (SIC) codes. The MIT Alumni database uses SIC codes to classify the industries in which alumni work.

These problems also fall under the category of weak equivalence. Though they could be resolved by IDT's, it is recommended that the problems be resolved by simply adopting the same set of codes in all databases concerned. It is recommended that all databases use Function Codes 1-19 and SIC Codes.

Chapter Five

Conclusions and Future Work

5.1 Conclusions

This study has demonstrated that a small, committed team who are familiar with the application can quickly implement an acceptable prototype system. It shows the effectiveness of informal communications in a loosely coupled organization such as a university. It suggests that calling for voluntarism can be effective in the management of such a group, but that effective monitoring, written communications, and interim deadlines are crucial for the timely production of a high-quality prototype. In addition, a project will incur a learning cost if unfamiliar software is being used.

Real data contains many ambiguities and semantic problems. Developers need to be aware of the tradeoff between simplicity and usefulness these problems imply, and be wary of extremes of simplicity or complexity.

5.2 Future Functionality

5.2.1 Future functions which are essential to the system

- a. Making the system secure. For example, it should be impossible for a student to see what priority cards another student has submitted. Job offer and salary information should also remain confidential. There is some information, however, that students should be able to share, for example jobs previously held. The SPO also maintains certain company information not shared with students, such as employer office contacts.
- b. Obtaining access to the actual MIT alumni database. This will involve getting the permission of the Alumni Office. In addition, it should be decided whether

the connection should be on-line or off-line (i.e. via a periodically updated tape).
If off-line, decide how often to update the data.

c. Other SPO functions such as company presentations scheduling, salary offers/
jobs accepted database, and list of companies ordering resume books.

5.2.2 Future functions which are central to the system, though not essential

a. Hardware acquisition for Placement Office / student use.

b. On-line resume book.

c. Correspondence opportunities database.

d. Making information available to recruiters (such as on-line resume book,
student job preference information), either at MIT or via dial-up.

5.2.3 Future functions which would be desirable, if technical, monetary, and data entry problems could be solved

a. Further work is needed to implement the solutions to the logical connectivity
problems discussed in Chapter 4.

b. Integrating outside databases such as Reuters and I.P. Sharp, for text and
financial performance information.

c. Database of printed materials available in Placement Office library.

d. Company / industry data on CD-ROM.

e. A system to aid the Placement Office in keeping track of the causal
relationships leading to success in students' job search. Why do only 60% of
students get jobs through on-campus recruiting, while 85% of students seek jobs
through this channel? The system would aim to track and analyze data to answer
this question.

f. Student access to system from their homes.

g. An intelligent help system that assists users in querying the database based on the observed competence of the user in using the system. This could be implemented using the heuristic features of CIS/TK.

Appendix A

PAS: A Sample Session

This sample session illustrates some of the most important functions of the system. It takes a student from signing onto the system, through looking at the recruiting schedule two ways (by location and by industry), submitting priority cards and checking which priority cards he has submitted, then signing off the system. Throughout, **boldface** indicates entries by the student.

The student begins by signing on to the system. The Main Menu (Figure A.1) comes up automatically.

Welcome to PAS; please indicate your USER STATUS below.

1. Student
2. Placement Office Staff
3. System Administrator

Use space bar, arrow keys, or type number to make selection.
Enter 'e' to return to previous menu or exit.
Enter carriage return to execute selection: **1**

Figure A.1: Main Menu

Since this is a student, he selects choice 1 and the Student Functions Menu is displayed (Figure A.2).

He selects Review Recruiting Schedule. The Recruiting Schedule Menu (Figure A.3) shows the various ways the student can select and list recruiting

Student Functions

1. Review Recruiting Schedule
2. Prepare Priority Cards
3. Use Programmed Aids for Job Selection

Use space bar, arrow keys, or type number to make selection.
Enter 'e' to return to previous menu or exit.
Enter carriage return to execute selection: 1

Figure A.2: Student Functions Menu

Options for Listing Recruiting Schedule

1. Chronological
2. Geographical
3. Company
4. Functional Area
5. Industry
6. Enter your own RSQL statement

Use space bar, arrow keys, or type number to make selection.
Enter 'e' to return to previous menu or exit.
Enter carriage return to execute selection: 2

Figure A.3: Recruiting Schedule Menu

schedule information. Let us assume our student is interested in jobs in Ohio. He selects 2 (Geographical) on the Recruiting Schedule Menu. The system responds with "What state are you interested in?" (Figure A.4).

The student types in the two-letter state abbreviation, in this case "OH." The

What state are you interested in? (Enter 2-letter code): OH

Recruiting Schedule by Location

State	City	Pos#	Title	Company	Date
OH	Cincinnati	40	Financial Management	Procter & Gamble	Feb 08
OH	Cleveland	6	Production Mgr	Standard Oil	Jan 26
OH	Cleveland	20	Business Analyst	BP America	Jan 26
OH	Cleveland	20	Business Analyst	BP America	Jan 27

Press Return to continue

Figure A.4: Recruiting Schedule by Location

system responds with a list of all recruiting positions in Ohio, sorted by city. The student notes the position number(s) of any positions that interest him.

Now let us imagine our student is also interested in the banking industry. Pressing *Return* after the display of Figure A.4 takes him back to the Recruiting Schedule Menu (Figure A.3). He chooses option 5 (Industry). The system responds, "Enter industry, surrounded by %." (The reason for the use of % is to work with the SQL "like" string searching capability). The system searches the recruiting firms by Standard Industrial Classification (SIC) code. It uses the SIC code to look up the industry in the SIC code table, and compares the name of the industry with the string the student entered. The "like" feature removes the need for the student to know the exact name of the industry. (In this case, the full name is Commercial Banking.)

The system displays the results of the student's query (Figure A.5). Firms

Enter industry, surrounded by % (e.g. %Equip%): **%Banking%**

Firms Recruiting From %Banking% Industry

Pos#	Company	Title	Date
15	American Express	Marketing Manager	Feb 10
16	American Express	Manager, Credit Risk	Feb 10
7	Citicorp	Corporate Finance Associate	Jan 27
22	Citicorp	Treasury Analyst	Jan 22
23	Citicorp	Management Associate	Jan 22

Press Return to continue

Figure A.5: Recruiting Schedule by Industry

from the Banking industry are listed, in alphabetical order by company. Again, if any of these positions interest him he notes the relevant position numbers.

After the student presses *Return*, the Recruiting Schedule Menu (Figure A.3) is again displayed. Pressing “e” returns the student to the Student Functions Menu (Figure A.2). This time, he chooses option 2 in order to submit priority cards. The Priority Card System Menu (Figure A.6) is displayed.

He here enters the number 1 to prepare priority cards. The Priority Card Entry Screen is displayed (Figure A.7). Since he wants to enter new priority cards, he presses “a” (for Add). The cursor goes to the “priority number” field. Here he enters the priority (1 to 15, where 1 is highest) he is placing on this position. Next, the student enters his social security number and the number of this position. If the student wishes, he can get position information on-line on this screen. He needs to press “t” (Table), which makes the next table on the screen (the Company table) active. He can then do queries to find companies by name or SIC code. Using the Company ID found via this query, he can then query

Priority Card System

1. Prepare Priority Cards
2. Report Priority Cards Submitted
3. Report Results of Interview Selection

Use space bar, arrow keys, or type number to make selection.
 Enter 'e' to return to previous menu or exit.
 Enter carriage return to execute selection: 1

Figure A.6: Priority Card System Menu

Query Next Previous Add Update Remove Table Screen Current Master
 Detail Output Exit ** 1: priority table**

priority number: [] student ssn: [] Position number: []

You are not allowed to change any of the information below.

Company information:	Company name:
	Company ID:
	SIC number:
Position information:	Company ID:
	Position number:
	Position title:
	Country:
	State:
	City:
Interview information:	Position number:
	Interview date:
	Room number:
	Interviewer:

Figure A.7: Priority Card Entry Screen

the Position table on this screen to find the relevant position number(s) and other position information. Once he has the position number, he can return to the top

of the screen (cycling through the tables by pressing "t") to enter in his priority card.

To end his PAS session, the student wants to check what priority cards he has submitted. He presses "e" to exit the Priority Card Entry Screen, whereupon the system responds with the Priority Card System Menu (Figure A.6). This time he chooses choice 2, Report Priority Cards Submitted. The system responds by asking for the student's last name (Figure A.8).

Please enter your last name: **Thomas**

These are the priority cards you have submitted:

Pos#	Title	Company	Date	Prio.
10	Corporate Planning	Ford Motor	Jan 28	4
12	Marketing Analyst	Merck	Jan 29	7
3	Associate Consultant	Touche Ross	Jan 25	8
2	Product Manager/ Program Mgr	Microsoft	Jan 25	10
6	Production Mgr	Standard Oil	Jan 26	15

Press Return to continue

Figure A.8: Report of Priority Cards Submitted

After searching the database, it displays the priority cards which Mr. Thomas has submitted. He can easily get a hard copy of this screen by pressing the Shift-Print Screen keys on the IBM PC.

After pressing *Return*, the student presses "e" (exit) three times to end the session.

Appendix B

PAS System Specifications

B.1 Tables in Recruit DB

Figure B.1 depicts the relationships among the important tables in RecruitDB.

*Denotes key field(s) for table

Name in (parentheses) is name on system.

1. Company (company)

command file company.sql
form ecom

Column Name	Type, Nulls	Comments
company__id*	char(8) not null	Unique alphanumeric ID
company__name	char(25) not null	Address of company headquarters
street__address	char(25) not null	
city	char(15) not null	
state	char(2)	
zipcode	char(5)	
country	char(15)	null if USA
sic__code	smallint not null	
employees	integer	Number of Employees
sales__assets	money(16,2)	Annual Sales / Assets (\$million)
parent__co	char(8)	Parent Company ID
Suggested field to add:		
resume__bk	char(1)	Purchased Resume Book (y/n)

2. Position (position)

command file position.sql
form epos

Column Name	Type, Nulls	Comments
position__num*	serial not null	Position numbers (1,2,3..) are assigned each year as job descriptions are received.
company__id	char(8) not null	Division name
position__title	char(35) not null	
division	char(20)	
contact__name	char(20) not null	
contact__title	char(20) not null	
contact__address	char(20) not null	
city	char(15) not null	

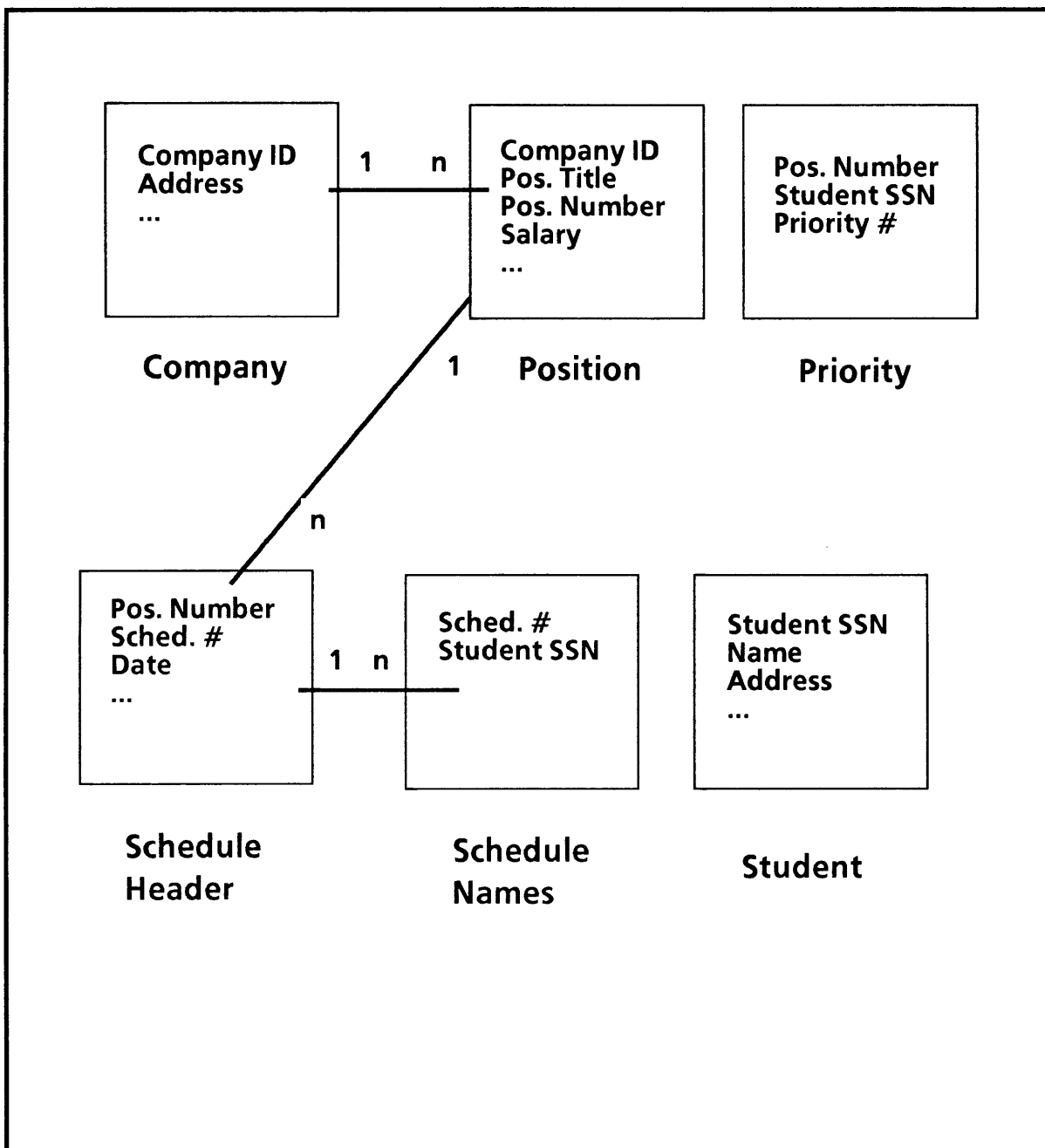


Figure B.1: The Important Tables in RecruitDB

state	char(2)	
zipcode	char(5)	
country	char(12)	null if USA
telephone	char(12) not null	
perm_summer	char(1) not null	Permanent / Summer (p/s)
cover_letter	char(1)	Cover Letter Reqd? (y/n)
intvw_length	smallint not null	Interview Length (min.)
num_hiring	smallint	Number of SM/MBA's to be hired

sal_desc	char(12)	Salary Description (e.g. competitive)
sal_low	money(8,2)	Base Salary range (low)
sal_hi	money(8,2)	Base Salary range (high)
ann_mthly	char(1)	Salary Annual/Monthly (a/m)
yeb	char(1)	Year End Bonus? (y/n)
stk_opt	char(1)	Stock Options (y/n)
profit_sh	char(1)	Profit Sharing (y/n)
citizenship	smallint	Citizenship Requmts (code)

Items not in Position table that are entered when creating positions:

- # of Open Schedules (Create one row in schedule header table for each open schedule)
- # of Closed Schedules (Create one row in schedule header table for each closed schedule)
- Recruiting date(s) (These go in Schedule headers)
- Function Code(s) for position (These go in Position Functions table)
- Location Code(s) (These go in Locations table)

3. Position Functions

Column Name	Type, Nulls	Comments
Position Number*	number	
Function Code*	number	Use codes 1-19

4. Position Locations

Column Name	Type, Nulls	Comments
Position Number*	number	
Location Code*	number	

5. SIC Codes (siccodb)

Column Name	Type, Nulls	Comments
sic_code*	smallint not null	Should sort alphabetically
industry	char(20) not null	

6. Schedule Header (sch_header)

command file schh.sql

One row per schedule. A schedule is typically twelve 30-minute interviews. (Other possibilities: eight 45-min. interviews or six 60-min. interviews.)

Column Name	Type, Nulls	Comments
schedule_num*	serial(1) not null	Unique 1,2,3.. each year
position_num	smallint not null	
intvw_date	date not null	
c_o	char(1) not null	Closed / Open / Both (c/o/b)
room_num	char(7)	Where interviews will be held

interviewer	char(20)	Interviewer's name
-------------	----------	--------------------

7. Schedule Names (sch__names)

command file schd.sql

One row per interview slot

Column Name	Type, Nulls	Comments
sched__num*	smallint not null	Schedule Number
time*	char(5) not null	Time of interview
student__ssn	char(11)	Student SSN

8. Priority Info. (priority)

(Up to 15 rows per student)

Column Name	Type, Nulls	Comments
student__ssn*	char(11)	Student SSN
position__num	integer	
priority__num*	smallint	1-15

9. Student (student)

One row per student

Column Name	Type, Nulls	Comments
student__ssn*	char(11)	Student SSN
first__name	char(15)	Student First Name
last__name	char(20)	Student Last Name
sm__expected	char(5)	Sloan SM Expected (year/month)
day__phone	char(8)	Daytime Phone#
eve__phone	char(8)	Evening Phone#
citizenship	smallint	Citizenship code (1 = US, etc.)
years__exp	smallint	Years Full-Time Work Exp.
ug__degree	char(4)	Undergraduate Degree
ug__major	char(20)	Undergraduate Major
ug__year	smallint	Undergrad. Year
grad__degree	char(4)	Grad. Degree
grad__major	char(20)	Grad. Major
grad__deg__year	smallint	Grad. Degree Year

10. Student Experience Details

One row for each job student has held

Column Name	Type, Nulls	Comments
Student SSN*	char	
Company ID*	char	
Position Title*	char	
Years	number	
Job Function Code	number	

11. Student Concentration Info (multiple rows per student)

Column Name	Type, Nulls	Comments
Student SSN*	char	
Sloan Concentration Code*	number	number, 1-14

12. Student's Preferred Job Function Info. (multiple rows per student)

Column Name	Type, Nulls	Comments
Student SSN*	char	
Function Code*	number	1-19

13. Student's Preferred Industry Info. (multiple rows per student)

Column Name	Type, Nulls	Comments
Student SSN*	char	
SIC Code*	number	

14. Student's Preferred Location Info.

Column Name	Type, Nulls	Comments
Student SSN*	char	
Preferred Location*	smallint	Location Code

15. Student's Preferred Firm Size

Column Name	Type, Nulls	Comments
Student SSN*	char	
Preferred Size	smallint	Code

16. Company Presentation Info.

Column Name	Type, Nulls	Comments
Company ID*	char	
Division*	char	
Presn Date	date	
Time	char	
Room	char	

B.2 Overall Logic**B.2.1 The Priority Card System**

Present system:

1. SPO releases list of all recruiting companies. It includes company name, date, number of schedules, closed/open/both, and position title. Students inspect that list and the job descriptions.

2. System operates on a one-week cycle. Generally students have a period of four days (Monday through Thursday) to submit cards for interviews to be held two weeks later. On Friday, the SPO sorts cards and posts a list of interview winners. Students winning interviews sign up for time slots on Monday and

Tuesday of the following week (one week before the interviews). A lottery (random drawing) is held on Wednesday to fill any remaining interview slots.

3. During the period described above, student submits cards for specific positions. The card can be any number from 1 (highest priority) through 15 (lowest priority). The student receives only one set of 1-15 for the year.

Automated system:

1. Interview lists can be generated from tables in Recruit DB. Can select by date range (of interviews) and select only open and half-open schedules. Students can use Recruit DB to search for appropriate positions.

2. Priorities Update (= card submission): Student inputs SSN, position # and priority #. System verifies that student has not already submitted this priority #. Information is stored in Priority Info table (see table descriptions, above). The word "card" will be used to refer to one row in the Priority Info table.

3. Priorities Sorting (Weekly function, performed by SPO):

a. Check student's citizenship and degree against job requirements. Possibly disqualify this card.

b. Remaining cards: sort by position #, priority #.

c. How many open schedules for this position?

Winners are top 12 (if 1 schedule)

top 24 (if 2 schedules)

top 6 (if 1/2 schedule), etc.

d. Report winners for students' information.

e. Report all cards submitted, for SPO information. If a space on schedule later becomes available, it will be filled from this list (in priority order).

Appendix C
Daily PAS Updates

To: The Group
 From: Larry
 Date: 5 April 1988
 Subject: Meeting minutes / Work plan / Daily PAS Update

This is my understanding of what was decided at our meeting this morning, plus a few afterthoughts. Please get back to me if your thoughts and mine don't correspond.

Andrea: Integrated Schema (a.k.a. Global Schema) for Placement Office / System

Remarks: It is desirable to have a rough schema as soon as possible, since the global schema is a parameter of CIS/TK. CIS/TK uses the global schema to "find" data in the underlying databases.
 Deliverable: (4/12): A draft schema diagram using Entity-Relationship model (cf. last term's final exam and Appendix 2 of Batini paper). The schema can be revised as we go along.

Andrea: Updated list of Sloan Placement Office (SPO) Documents

Remarks: All my SPO documents are in the blue looseleaf in my locker.

Lisa: A report on the organizational issues (5-10 pages double spaced)

Remarks: Report should mention our student survey, meetings with Linda, etc. Andrea, Drew and I will share with Lisa any information about and/or experiences we have with the organizational issues.
 Deliverable: (4/12): Outline or draft of report

Drew: User Interface

Remarks: I spoke with Rich about this. He agrees on the importance of a good user interface. And following Stu's "not just a footprint in the sand" philosophy, he agrees that a working user interface is much to be preferred to a paper one. Therefore, we are aiming to implement this month a working user interface for a single-database system. Since our version of Oracle doesn't seem to include menu-designing capabilities, the best idea is to put the RecruitDB on the 3B2 and design the user interface using Informix's FORMBUILD screen form generator. Our philosophy will be:

Development Machine: 3B2 (or RT), only one database

Target "Machine": KOREL, multiple databases

We will use the Development Machine to test out ideas and gain acceptance for the system.

Deliverable: (4/12): Preliminary user interface design and timetable for completion. To quote Rich, "If Drew does a good job, the user interface becomes a blueprint for future systems."

(continued)

Larry:

- Completing the table specifications for the RecruitDB
- Completing the overall design plan, in priority order
- Completing revised thesis outline
- Entering test data
- Planning future capabilities / hardware (with Lisa's help)
- Coordinating the group
- Designing 3 to 5 sample queries and specifying how each one would use the databases and use semantic reconciliation and CIS/TK
- Designing CIS/TK applications for PAS (e.g. rule systems)
- Designing object models for PAS
- Documenting the project and our methodology
- Preparing presentation for Linda (with help from group)
- Acting as liaison with TK, Alec & Tom
- Reporting to Rich Wang, Stu Madnick, and Dave Rogers

Deliverable: (4/12):

Completed RecruitDB table specifications

Some test data

Completed design plan

Revised Thesis Outline

Example queries, as described above

Suggestions on presentation for Linda

Preliminary CIS/TK applications / rules

Preliminary object models

Draft of future (1988-89) system

To: All Members of the Loop
From: Larry
Date: 13 April 1988
Subject: Daily PAS Update

Status

Attached are updated (but not final) versions of:

- PAS Design Plan
- Scenarios
- Table Definitions

I have begun creating the tables on MIT2C.

The other group members are in the process of creating:

Lisa	Report on Organizational Issues/Future Issues
Andrea	Global Schema
Drew	User Interface, for Students and Placement Office

Rule System

I have tentatively decided to create a rule system to suggest companies / positions based on area of concentration and expressed preferences such as locations. This is along the lines of what we discussed at today's meeting. I would be grateful for any suggestions/help on this.

Another idea to explore (for next version?) is Dave's idea of a "help" system that assisted users in querying the database "based on the observed competence of the user." This looks promising for a rule system. Comments?

Daily PAS Update

19 April 1988

Demo Tentatively Scheduled

Rich has suggested that we demo the system for the 15.579 class on Tuesday, April 26. I think this is a good idea, as it will give us an interim goal to shoot for. I'd like to have up a reasonably working version of at least the Priority Card system by that date, with, hopefully, a decent user interface. I'm really going to need your help on this so I'd like to discuss on Thursday what needs to be done.

More Progress Made on System

I am attaching the latest versions of:

- Recruit DB Specs (Tables, Connectivity problems, Priority Card system)
- Organizational / Strategic Blurting (Meant to be read in conjunction with

Lisa's report)

As usual, I would welcome any feedback or comments.

Daily PAS Update

29 April 1988

Whew. This has been one exciting week, PAS people.

Suggestions from Prof. Madnick

On Thursday (April 28) I demonstrated the system for Stu Madnick. Following are some of his criticisms and suggestions, with which I agree.

General

1. The system is slow.
2. In general, the system is difficult to understand. We need to pay more attention to human factors. One good idea would be just to take any random student, sit them down in front of the machine and let them loose. Whatever they can't understand or complain about is a candidate for change.
3. Output of select statements, if more than one screen, just whizzes by. It would be preferable if: a) It were in columnar form, and b) It paused between screens.
4. We should enter a bit more data, to make it easier to test and to demonstrate features.

Specific

1. The screen for entering dates is difficult to understand.
2. The "priority cards submitted" report doesn't make it clear what student this is for.
3. Geographical recruiting report - it would be nice if the user could specify a geographical area.
4. In "Review Recruiting Schedules" it would be nice to be able to ask if a specific company was coming, such as IBM. I.e. By Company -> "Enter name or hit return for all," and the "Enter name" should support "like" (String search) to avoid problems with different spellings.
5. It is difficult to figure out how to get information on the Priority Card Entry screen.

We should try to get as many of these problems as possible resolved before Tuesday (demo for Linda Stantial). Some are a lot easier than others.

We have scheduled a work session for Monday evening, May 2, to improve the system.

Bibliography

[BAT86] Batini, C., M. Lenzerini, and S.B. Navathe, "A Comparative Analysis of Methodologies for Database Schema Integration." ACM Computing Surveys, 18, No. 4 (December 1986), 323-364.

[BEN88] Benjamin, Robert I., David W. DeLong, and Michael S. Scott Morton. "The Realities of Electronic Data Interchange: How Much Competitive Advantage?" CISR Working Paper No. 166, M.I.T. 1988.

[BRO75] Brooks, Frederick P. The Mythical Man-Month: Essays on Software Engineering. Reading, MA: Addison-Wesley, 1975.

[ELM87] Elmasri, R., J. Larson, and S. Navathe. "Schema Integration Algorithms for Federated Databases and Logical Database Design." Submitted for Publication, 1987.

[HOR88a] Horton, David C., et al. "The Design and Implementation of the CIS/TK Query Processor Architecture." CIS Project Report #CIS-88-02, M.I.T. 1988.

[HOR88b] Horton, David C., et al. "Mapping Data Between Different Representation Schemes in CIS/TK." CIS Project Report #CIS-88-03, M.I.T. 1988.

[KID81] Kidder, Tracy. The Soul of a New Machine. Boston: Little, Brown, 1981.

[LEV87] Levine, Samuel P. "Interfacing Objects and Databases." Master's Thesis, M.I.T. 1987.

[MAD87] Madnick, Stuart E. "Information Systems and Services: Vision Statement." Manuscript, M.I.T. 1987.

[MAD88a] Madnick, Stuart E. and Y. Richard Wang. "Facilitating Connectivity in Composite Information Systems." To appear in Data Base.

[MAD88b] Madnick, Stuart E. and Y. Richard Wang. "A Framework of Composite Information Systems for Strategic Advantage." In Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, Vol. III, January 1988, pp. 35-43.

[MAD88c] Madnick, Stuart E. and Y. Richard Wang. "Integrating Disparate Databases for Composite Answers." In Proceedings of the Twenty-first Annual Hawaii International Conference on System Sciences, Vol. II, January 1988, pp. 583-592.

[MAD88d] Madnick, Stuart E. and Y. Richard Wang. "A Tool Kit for Composite Information Systems: Research Overview, Current Status, & Near-Term Plan." CIS Project Report #CIS-88-01, M.I.T. 1988.

[MAS87a] Massachusetts Institute of Technology Sloan School of Management, 1987 Placement Manual.

[MAS87b] Massachusetts Institute of Technology Sloan School of Management, 1987 Placement Report.

[MIT88] "M.I.T. School of Management Masters Program Outline." Manuscript, M.I.T. 1988.

[OSB87] Osborn, Charley. "Towards a CIS Model for Strategic Applications." Knowledge Based Integrated Information Systems Engineering Project Technical Report #22, M.I.T. 1987.

[SCH83] Schön, D.A. "Organizational Learning." In Beyond Method: Strategies for Social Research, Gareth Morgan, ed., Sage, 1983.

[TEN88] Tener, Lisa. CIS Project Report #CIS-88-06, M.I.T. 1988.

[TRI87] Trice, Andrew, et al. "Placement Office Requirements." Manuscript, M.I.T. 1987.

[WEI76] Weick, Karl E. "Educational Organizations as Loosely Coupled Systems." Administrative Science Quarterly, 21 (March 1976), 1-19.