Indiana University

The University Graduate School

Pilot Project in Electronic Application

Final Report

Eric Jeschke
Computer Science Department
jeschke@cs.indiana.edu

June 2, 1995

Contents

1 Introduction **1**
1.1 History . . . 1
1.2 The Approach . . . 1
1.3 Organization of this Report . . . 2

2 The Paper Application System **2**

3 The Electronic Application System **4**
3.1 Design Considerations . . . 5
3.1.1 Choosing a Medium . . . 5
3.1.2 Security and Authentication . . . 6
3.1.3 Signatures and Liability . . . 6
3.1.4 Transcripts and GRE Scores . . . 7
3.1.5 Fee Payment . . . 7
3.1.6 Archival . . . 8
3.1.7 Choice of Tools . . . 8
3.2 Implementation . . . 8

4 Experience with Electronic Application **10**

5 Recommendations **13**
5.1 Third-party Options . . . 13
5.2 University Computing Services . . . 13
5.3 Creating a Production Web Application System . . . 14
5.3.1 The RUGS Web/Database Server . . . 15

List of Figures

1 Graduate Application at IU . . . 3
2 The Electronic Application Prototype System . . . 9
3 The IU Electronic Application Home Page . . . 11
4 Second Generation Electronic Application . . . 17
5 The Signature Form . . . 19

1 Introduction

Indiana University's Pilot Project in Electronic Application was designed to explore the feasibility of on-line electronic application to IU's graduate programs. The ultimate goals of electronic application are similar to those envisioned for many paper-to-electronic forms processing initiatives:

- Reduce materials, printing, mailing, data entry, personnel, and other costs associated with printed applications.
- Reduce physical storage space requirements for current and archived applications, while providing more efficient data retrieval.
- Reduce application processing time and overhead.
- Increase the visibility and accessibility of IU's graduate programs, and provide a competitive edge to IU in terms of graduate application.

1.1 History

The project was initiated in the spring of 1993 by Frank Hoole, Associate Dean of Research and the University Graduate School (*RUGS*), in consultation with Steve Johnson, Assistant Professor of Computer Science and Chair of the department. The two put together a proposal to fund a graduate assistantship for a pilot study of electronic application at IU. Funding for the project was approved by George Walker, Vice President for Research and Dean of the University Graduate School. Eric Jeschke, a doctoral student in the Computer Science Department, was appointed to begin the assistantship in June 1993. Kim Bunch, Information Manager at *RUGS*, would provide technical assistance and oversight at the graduate school.

1.2 The Approach

The pilot project had the following approach and objectives:

1. Analyze the current paper application system to determine the information flow and requirements for the proposed electronic system.
2. Evaluate a variety of electronic transmission and delivery alternatives for all aspects of application: forms, letters of recommendation, transcripts, GRE scores, and fee payment.
3. Develop a prototype system capable of accepting electronic applications.
4. Test and refine the system over the course of several semesters to gain useful experience and feedback on the electronic application process from both end-users and administrators.

The prototype system would be limited to prospective Computer Science students, and further, to domestic applicants only, since other admissions would require special processing that was outside the project's scope and jurisdiction. However, the project would be developed with the idea of eventually incorporating other technologically-capable departments, possibly within the second year. In addition, the project would have the following guidelines:

- Minimize disruption of current work flow and information channels; the electronic system would have to coexist with the existing paper system and not interfere or add significant complexity to the system.
- The project would have minimal expenditure. Funding for the project consisted solely of a half-time (.50 FTE) graduate assistantship to develop the project. The hardware, software and electronic infrastructure would be limited to what was available within the Computer Science Department and Graduate School.

1.3 Organization of this Report

This report summarizes the results and findings of the electronic application project after two years of design, development and use. First we describe the current paper application system at IU, which gives an overall view of the flow of application information within the university. We then describe the design of the prototype electronic application system and our experience using the system for actual admissions over the course of four semesters. We conclude with a summary of the project's current status and recommendations regarding the future of electronic application at IU.

2 The Paper Application System

IU's graduate application procedure combines a distributed admissions process with centralized record-keeping. Application packets are mailed directly from individual departments to requesting applicants. Each application packet contains a standardized, optically-scanned bubble-form application, any auxiliary application forms required by the department itself, informational material, and an envelope addressed to the bursar for payment of application fees. The *RUGS* bubble-form includes sections for personal data (addressing and statistical information), application data (e.g. educational history), and state residency information. Additional application requirements usually include at least three letters of recommendation, a statement of purpose, transcripts from all undergraduate institutions attended and official Graduate Record Examination (GRE) scores submitted by the Educational Testing Service.

Figure 1 depicts the handling of applications submitted to the Computer Science department. The work and information flow is representative of many departments on the Bloomington campus. The application materials are received and processed by our department's graduate secretary, Pam

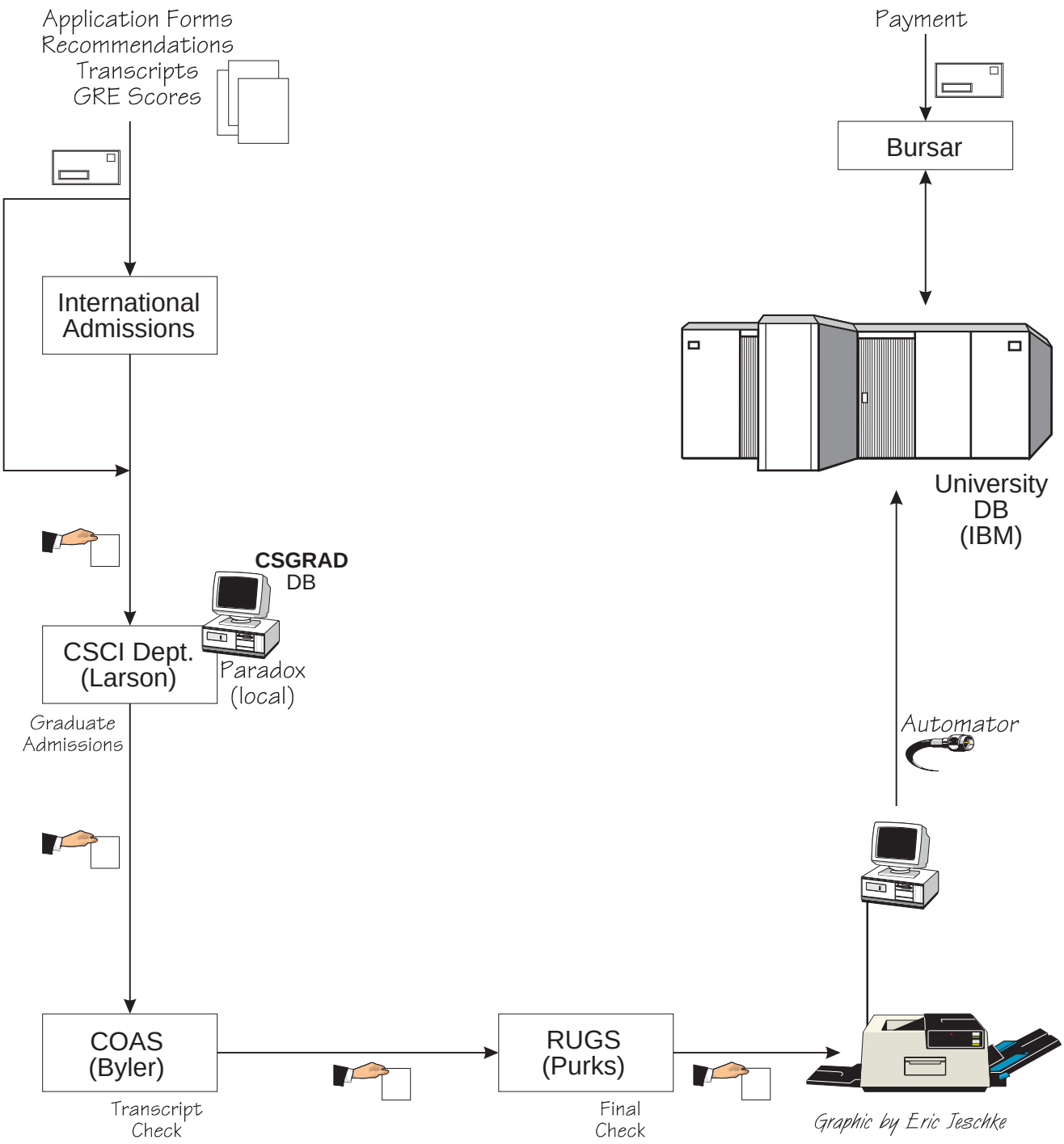

Figure 1: Graduate Application at IU

Larson. Additionally, non-domestic applications (i.e. other than from U.S. citizens and permanent residents) are first processed by the office of International Admissions, which assembles the applications and routes them to Computer Science when complete.

Like many departments, Computer Science uses a local database system to manage their graduate applications. Pam Larson enters the information from the paper applications manually into the database. The *CSGRAD* database contains most of the information required by the graduate school, plus some specific information useful for the departments admissions and award decisions, such as grades received in prerequisite computer science courses, computer language skills, the dates certain documents were received, etc. In addition to tracking applications, the database provides a way to generate statistical data on applicants, rank applicants according to several criteria, generate reports, and serves other miscellaneous uses. The *CSGRAD* database is implemented in Borland's *Paradox*, a desktop database product for PC's, and the standard used by the CS office staff.

Pam prepares any reports required by the admissions and awards committee. After the committee has made admission decisions, the standardized bubble-forms for accepted students are routed with transcript and GRE information to the graduate school. The information is reviewed twice before computer processing. Mitchell Byler, Assistant Dean in the College of Arts and Sciences, verifies the information provided on the application form against the official transcripts and GRE scores. Major omissions or inconsistencies might result in the forms being returned to the department for correction. Carol Purks, Admissions Coordinator, and other Graduate School staff perform final checks for completeness before processing the forms through an optical scanner.

Several applications can be successively scanned, producing a file on the PC attached to the scanner. The information in the file is uploaded to the university's mainframe databases via an automated interface customized by IU's University Computing Services. This upload establishes official IU records for the student with the Bursar, Registrar, and other university entities. The processed bubble forms are archived at the graduate school for a period of three years.

3 The Electronic Application System

The *RUGS* scanner-based automation is only about three years old. Until its introduction, applications were entered manually into the IU system by the staff at the graduate school. The optical scanning procedure advanced the level of automation in that process by drastically reducing manual data entry once the applications reached *RUGS*. This streamlined process allows the graduate school to handle a large number of university-wide graduate applications with a staff of 2-3 persons. However, because of the distributed nature of graduate application, manual data entry is performed by departments all over the university into local databases like *CSGRAD* before the applications ever get to *RUGS*. One of the primary goals of electronic application is to push this manual data entry all the way out to the applicant, who already performs it in the process of filling out the application. The application data flows through the same administrative channels,

but is processed and transferred electronically between the applicant, department, and offices of the graduate school. Thus, our project is designed to mirror the existing paper work-flow at IU electronically, while preserving the current system and disrupting it as little as possible. This preserves the existing administrative infrastructure (for better or worse) and requires no organizational changes, just technological adaptation.

3.1 Design Considerations

A number of issues arise in the transition from paper to electronic forms processing that require special consideration. These issues had to be resolved in the design of the electronic application system at IU.

3.1.1 Choosing a Medium

The first variable to be determined was the transmission medium for electronic applications. Two strong possibilities emerged from a short list of alternatives:

1. Distribute the application as an information gathering program on floppy disk, which would be mailed to the applicant with other informational materials, possibly as an addition to the regular application packet. The applicant would run the program on the floppy disk, and the program would present a series of graphical forms to be filled out. The information would be saved back to the diskette, to be mailed back to the department, or transmitted via modem.

2. Develop an Internet-based application which the applicant could fill out at their site and submit to the department remotely via e-mail or the World Wide Web.

Choice No. 1 would allow a sophisticated, completely customizable electronic application form to be developed. It could include context-sensitive help, immediate formatting and error corrections, etc. On the downside, although it would automate data entry of the incoming data, this method would not reduce mailing costs, and might, in fact, increase them. Additional drawbacks to this method are that it makes correcting form flaws and adding or deleting information difficult once the disks are written or distributed, and the difficulty of insuring hardware/software compatability for a variety of platforms (i.e. *DOS*, *Windows*, *Macintosh*, etc.).

Choice No. 2 would allow anyone with Internet access to apply electronically. Although this limits availability, most undergraduate schools as well as many corporations and government agencies already have Internet access. This implementation route eliminates many of the drawbacks associated with choice No. 1: there are no platform limitations, forms are more easily updated as needed, mailing costs decrease or remain at present levels, and it obviates the need for manual handling of diskettes. The biggest drawback with this method is that formatting, on-line help, and error correction are not as immediate, because processing of the data can only occur when the data is received by the department. Considering the two alternatives, the choice fell to the Internet approach.

The first prototype system was e-mail based, since e-mail is a ubiquitous lowest common denominator in terms of platforms and Internet services. Later, as the World Wide Web began to aggressively expand and browsers became available for all desktop platforms, the e-mail interface was abandoned in favor of a forms-based Web interface, which greatly reduced parsing and form layout problems.

3.1.2 Security and Authentication

Security and authentication are among the first issues that are broached in the context of electronic application. The main concerns are forgery, privacy, tampering, and other malicious acts. These concerns must certainly be addressed, but it is important to realize that many of these same problems are present in the existing paper application system, or for computer networks in general. We made a decision in the early stages of the project to not let the progress get bogged down by trying to address authentication and security issues that are outstanding problems for the existing IU application system, beyond what seems reasonably necessary to handle the transition to on-line application. As an example, consider that a student can easily forge a letter of recommendation, whether sent through U.S. mail or Internet e-mail. In either case, if the university has reason to suspect forgery, a simple check (e.g. by telephone, fax or registered U.S. mail) with the actual recommender to verify the letter's contents would suffice to determine its validity, irrespective of the actual method of transmission.

There are electronic (cryptographic) methods for securing and authenticating electronic documents; for example, *PGP* (Pretty Good Privacy) is a widely available public-key cryptography system. These methods are currently not integrated into most network mail and web software, and are cumbersome to use otherwise. This situation is expected to change as the Internet tools mature and Internet growth expands further into the area of commerce. In the interest of compatibility and simplicity, we decided not to employ cryptographic methods. This decision was also based on the observation that graduate applications, while subject to tampering and authentication concerns, are not in the same league as, say, financial transactions, and are therefore much less likely to be a target for malicious interference.

The bottom line is that we did not attempt to provide authentication *beyond* what is required and expected of the paper system, as these problems are beyond the scope of this project. Security is addressed through the use of password-protected application forms. An applicant can only access their data by supplying the proper password. Electronic letters of recommendation are also password protected; each recommender is given a separate password.

3.1.3 Signatures and Liability

An issue related to authentication is *liability*. Signatures are required on the paper application forms in at least five places. The signatures testify that the applicant has not falsified or knowingly misrepresented information about the information in the application or in their state residency

status, and whether they have waived access to the three letters of recommendation written on their behalf. These signatures help defend IU against potential lawsuits by the applicant on issues ranging from academic dismissal to fee payment to other unforeseen litigation.

Because a standard, universal, electronic authorization scheme has not yet emerged (see section above), we determined that a signature would be necessary for the reasons outlined above. We drafted a single statement on the three areas (application, residency, and recommendations). This statement was circulated through the IU legal department by *RUGS* to make sure that it adequately addressed the same issues as the paper application. A report is generated for each accepted electronic applicant; an example is appended to this document (page 19). The applicant must sign this form when they arrive on campus in order to receive a registration authorization slip. This is handled where the registration slips are distributed, typically by the graduate secretary of the department.

3.1.4 Transcripts and GRE Scores

Official transcripts and GRE scores are mailed from the issuing institution and the Educational Testing Service, respectively. There is an electronic interchange standard for transcripts, called *SPEEDE* (Standardization of Postsecondary Education Electronic Data Exchange). This standard is being actively developed by the American Association of Collegiate Registrars and Admissions Officers (AACRAO). Many universities, including IU, have not yet signed on to this evolving effort. It is beyond the scope and financial means of our pilot project to engage IU in this standard, which at this time requires connection to a proprietary *VAR* (Value Added Network) and commercial *EDI* (Electronic Data Interchange) software to handle the transmissions. Internet connectivity for *SPEEDE* is currently under investigation by AACRAO members.

GRE scores are available "electronically" on floppy disk or tape, distributed periodically by the Educational Testing Service (EDS), administrators of the exam. The graduate school has made these available through a *Paradox* interface on a server as a separate project. This system has not yet been cross-connected with the electronic application system, although this is would not be difficult. The latest GRE scores may not be available through that interface at the time that the new records are being processed.

Until such time as these documents can be received electronically they must be handled as a paper adjunct to the application system. Applicants provide their own GRE scores and educational history, which are validated against the official documents when they arrive via surface mail. These paper adjuncts are processed through the normal channels, as described in section 2.

3.1.5 Fee Payment

The office of the bursar directly handles application fee payments. We investigated the possibility of application fee payment by credit card over the telephone, as they are set up to handle these kind of transactions for enrolled students. However, the bursar seems reluctant to accept payments

for applicants this way, possibly because no records exist for them in the university computers. This situation can likely be rectified by an official policy change authorizing these payments by the bursar administration. Until then, payment must be mailed directly to the office of bursar.

3.1.6 Archival

Parts of the paper applications are archived at *RUGS* for a period of three years. *RUGS* can archive the electronic applications as well, by backing up the database to DAT tape and periodically purging records that are dated longer than the desired storage interval. Because the information can be stored in a much smaller amount of physical space, records can be archived for longer periods.

3.1.7 Choice of Tools

We attempted to integrate with existing systems and tools wherever possible; first, to facilitate ease of use and minimize disruption, and second, because the project had no funding to acquire additional hardware or software, which meant we had to use whatever tools were available on the platforms that were available. The office staff of the CS department as well as the staff of *RUGS* have standardized on *Paradox* as a database platform, so *Paradox* was a logical choice for the database component of the project. Unfortunately, *Paradox* runs under *DOS*, and *DOS* is not a stable platform for the TCP/IP networking used for a World Wide Web server. Additionally, the CS department's staff PC's are not equipped, hardware- or software-wise, to handle the task of web server. These limitations led us to chose a *Unix* workstation to handle the Web application service; the more robust TCP/IP networking plus abundance of good Web interface tools such as the *Perl* language aided this decision.

3.2 Implementation

Our prototype system uses the following design. The paper system is complemented by an on-line counterpart, as shown in figure 2. The application forms are implemented by a set of World Wide Web forms backed by *CGI* (Common Gateway Interface) scripts written in the *Perl* language. A *Unix* web server handles requests for the forms over the Internet. The Computer Science application home page (see figure 3) has a menu of hyperlinks from which one can submit a new application or access an existing one.

The application data is stored in *Unix* files on the server. Applicants can access their forms at any time by providing the proper password, which is generated when they elect to start an application. This allows the forms to be filled out piecemeal, at the applicant's leisure, and to correct mistakes or fill in information as it becomes available, much the way a paper application might be filled out. Letters of recommendation can also be submitted electronically and are stored with the applicant's data. This is handled by a special form that the recommender can access by supplying the proper applicant ID and per-recommender password. The URL and password

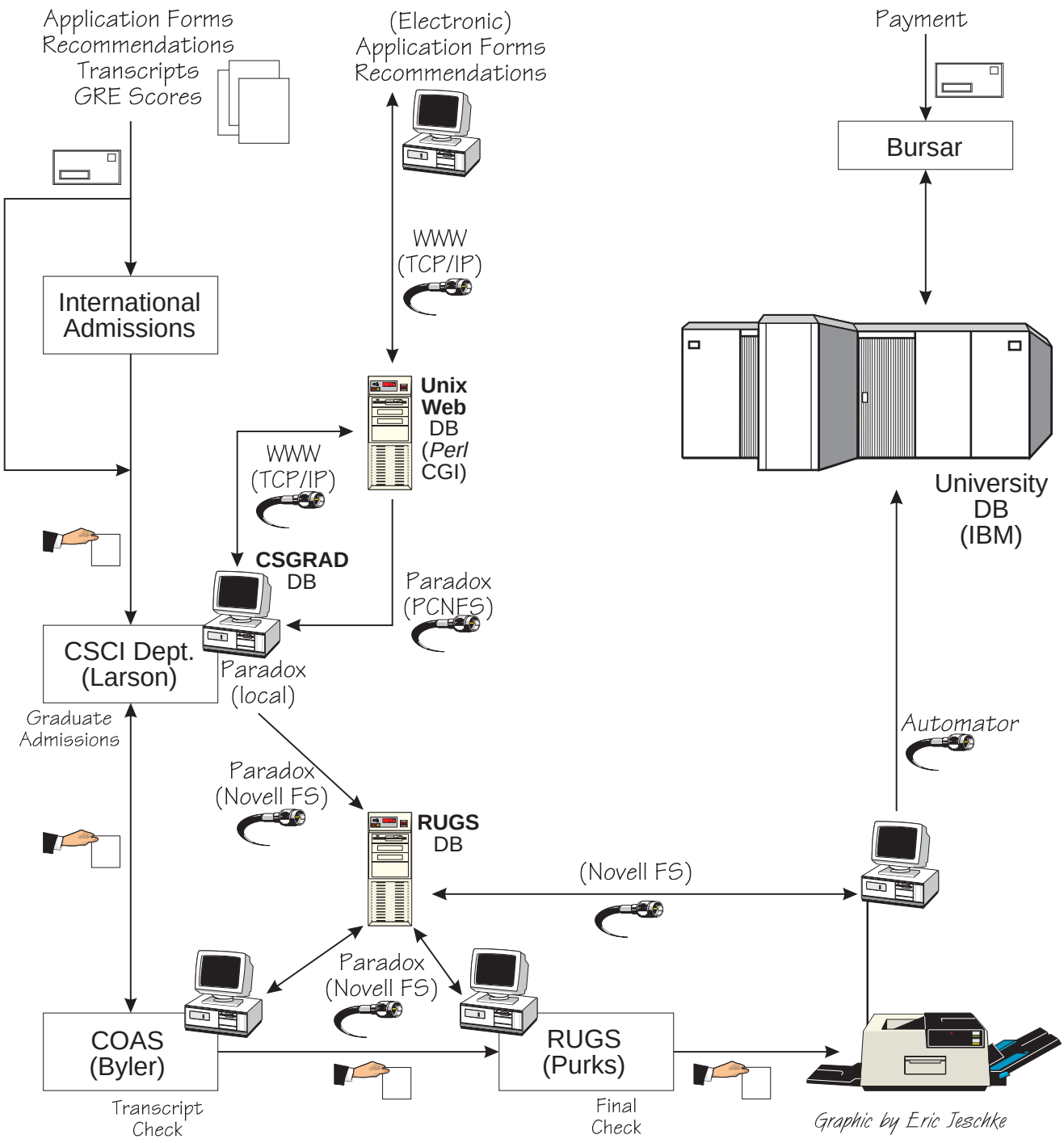

Figure 2: The Electronic Application Prototype System

are e-mailed to recommenders for which the applicant provides an e-mail address when submitting their application.

A private set of administrator Web forms allow Pam Larson to view the applications that have been submitted and download them to her *Paradox CSGRAD* database when they are complete. These forms also allow her to prepare a number of standard reports in *Postscript*, e-mail the applicant or recommenders, leave notes for the applicant to see when they check their status, and delete applications, all through the web browser interface.

Once in *CSGRAD*, the applications join the paper application records that were manually entered. From this database, Pam can manipulate the data in the same fashion as she can for the paper applications that she entered manually. After the admissions committee has finished its deliberations, the records for both electronic- and paper-based applications are transferred to a *RUGS* database for further processing. This transfer is handled by a *Paradox* script which copies the subset of information common to both databases. Since *RUGS* also uses *Paradox* as an office standard, the transfer is simpler than between the *Unix* database and *CSGRAD*, since it can be performed entirely within *Paradox* over a *Novell* PC network.

Once in the *RUGS* database, applications go through two final stages corresponding to the steps performed for paper applications. Mitchell Byler and Carol Purks perform final checks on the new records before they are uploaded to the university mainframes. The paper transcripts and GRE scores are sent over to the graduate school after the records have been transferred electronically. The *RUGS* database allows the new applicant records to be called up by one of several indexes, including name, social security number, and application number. Mitchell and Carol call up the records corresponding to the transcripts and review the records individually. The review process is nearly identical to that performed on the paper applications, except that it is performed on the computer. Special fields in the *RUGS* database record the dates and initials of the reviewers as they sign off on the forms.

After Carol Purks has approved a batch of records, she runs a *Paradox* script thats dumps any new records to a file compatible with that produced by the optical scanner on the standardized paper forms. The same *Automator* procedure that is used for the optically-scanned forms can then be used to upload the electronic applications to the main university databases.

4 Experience with Electronic Application

The prototype electronic application system has been online and accepting applications for over a year. Experience with the system has been generally positive. The system went online in late December of 1993, well into the Fall 94 application period. For that period we received 17 electronic applications out of 107 domestic applications, a 15% rate. For Fall of 95 we received 15 electronic applications out of 65 domestic applications, a 23% rate. There were no domestic applications for Spring 94.

Feedback from applicants and recommenders was generally favorable. There is a normal ("bell

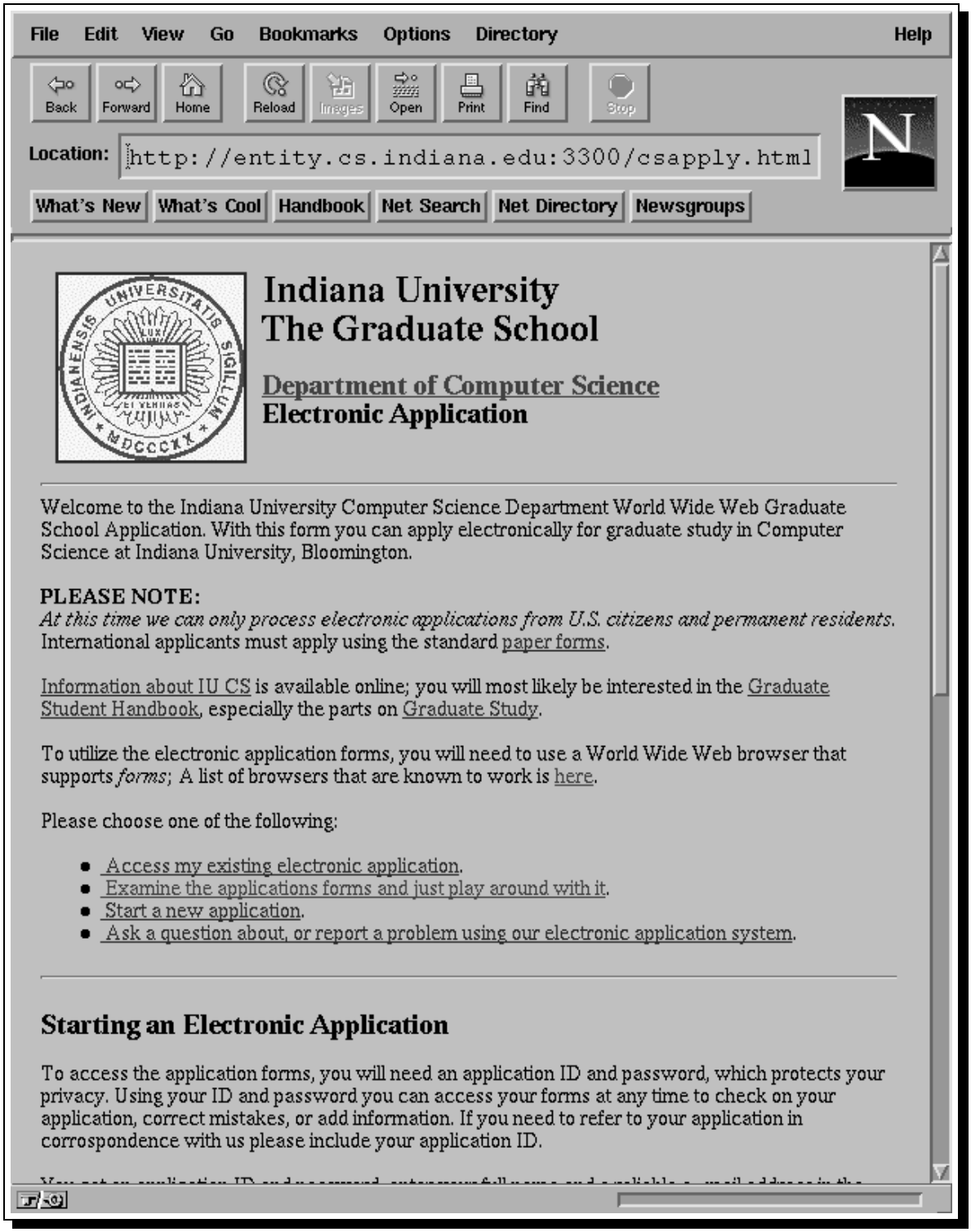

Figure 3: The IU Electronic Application Home Page

curve") distribution corresponding to when individuals will embrace a new technology. For electronic application, applicants for graduate study in Computer Science would have to be considered in the "early adopter" category as far as electronic application is concerned. Nevertheless, electronic application is such a new thing, and so few schools are actually doing it, that it is difficult to judge how good or bad our usage was compared to the paper application numbers, especially because of some of the other factors involved (no international applicants, late notice, etc.). Given that the bulk of Computer Science graduate applications are international, and considering the other factors, the number of electronic applications we received is a good sign that it is a system that definitely will be used.

Although we made a concerted effort to avoid disrupting the current workflow, the addition of a secondary system meant extra work for Pam Larson in the Computer Science department. Pam had to learn a new tool (Web browsers) to interact with an additional database of incoming electronic applications and manage this in addition to, and in conjunction with, the paper applications and the *Paradox* database with which she was familiar. The Web database contained application information such as letters of recommendation, statements of purpose and residency information that was not downloaded to *CSGRAD*, because of Paradox's limitations on field size and to avoid bloating the database. This information, ironically, had to be printed as reports to distribute to admissions committee members who were used to the paper documents for decision-making purposes, although the information can also be viewed online through the Web. Although web browsers present an easy to use interface, and we customized it for a number of ways to print and view the data, this new way of storing and manipulating data took getting used to, in contrast to the good old filing cabinet.

Some of the problems Pam faced can be traced to the use of separate databases for the web server, the departmental database, and the graduate school database. The reason separate databases were used is primarily due to *DOS*'s limitations for Web connectivity and Paradox's arcane limitations for a multi-user networked database. For example, *Paradox* would not allow us to keep the *RUGS* data in one (shared) table and the CS data in another, and combine them into a multi-page form. These platform limitations forced us to migrate data from one database to another, which generally meant additional administration overhead for Pam. Although we attempted to simplify and automate the process as much as possible, a certain amount of manual intervention and administration was unavoidable. An early experiment to unify the databases and sidestep these problems was unsuccessful. We attempted to use a *Unix Sybase* server to store the data and a *Paradox-to-Sybase* SQL Link to connect the two *Paradox* databases to the server, so that Pam could administer the server database from *Paradox*. However, connectivity between the PC client and *Unix* server was intermittent, and the capabilities for manipulating the data from *Paradox* were cumbersome and limited; the remote tables could not be manipulated with the same functionality and ease as local tables. For example, it was extremely difficult to add a field to, or restructure the remote table, or directly view, enter, modify or delete data through the standard *Paradox* interface, tasks which account for the bulk of Pam's interactions with the database. The Paradox SQL Link product additionally destabilized the DOS/Windows platform, and after a hard

crash resulted in some data loss the approach was abandoned in favor of the separate databases approach. Although this attempt was unsuccessful, I believe the client-server approach is the correct one to take, and with careful evaluation a suitable product can be found.

For Mitchell and Carol at the graduate school, the electronic application system integrated easily into the work flow there. This success is probably due to two factors. First, the nature of their transaction with each application (a review of data) is a brief, well-defined procedure. Adapting this from paper forms to computer screen forms is fairly straightforward, and the computer interface was well-received. Secondly, the *RUGS* component of the electronic application procedure used tools (*Paradox* and *Automator*) with which they were already familiar. The electronic system would temporarily displace the paper system when they used it, and did not result in extra work as it did for Pam; when they did use it, they were operating entirely in *Paradox*, a system in which they were comfortable.

5 Recommendations

We conclude our report with some recommendations for *RUGS* regarding commercial electronic application systems, the role of *UCS* in electronic application and the potential for scaling up the current prototype to handle applications for multiple departments.

5.1 Third-party Options

RUGS is currently considering the possibility of contracting out the electronic application process to a commercial service. A number of companies are beginning to tap into the electronic application market, especially at the undergraduate level. Most of these companies claim to support the emerging *SPEEDE* standard for academic record interchange, which is being enhanced to support electronic application as well as transcripts. Contracting out electronic application would insulate *RUGS* to a certain degree from trying to track evolving standards and technology in this area. On the other hand, *RUGS* should be careful about getting locked into a proprietary solution and should evaluate potential contractors on their technology, services, and level of integration with *RUGS* existing systems and long-term vision for graduate application procedure at IU.

5.2 University Computing Services

University Computing Services/Academic Computing may be considering developments in electronic application on behalf of the offices of registrar and bursar. There may also be a downsizing effort underway for the university mainframes, and that would also impact *RUGS* in this area. In particular, it might open up the possibility for tighter integration between Academic Computing, *RUGS*, and various departmental information systems based on client-server database interactions. The *FMS* (Financial Management System) project is similar to electronic application in terms of

department-university database interactions, and might serve as a good model and potential source of information. *RUGS* should consult with *UCS* regarding their long-range planning in this area.

5.3 Creating a Production Web Application System

The World Wide Web, already an established standard in electronic document dissemination, is also becoming a standard for electronic forms processing. Many Universities are setting up Web-based electronic application systems, both at the undergraduate and graduate level. IU is at the forefront in this regard, having already developed a working prototype. IU's system does not currently address national standardization efforts such as *SPEEDE*. Nevertheless, IU's system provides benefits in terms of control over content, appearance and functionality, and affords a competitive advantage to IU compared to universities that are adopting a wait-and-see attitude toward electronic application.

The project's main goal was to determine the issues and problems involved in the process of electronic application and to determine its feasibility within the current framework at IU. In this regard, the project has succeeded "in spades", having shown that it can indeed be done, and that people will use it. However, the current system *is* a prototype. It was put together rapidly, and at very low cost using available hardware and software components at Computer Science and *RUGS*. It is is not suitable, in its current incarnation, for scaling up to a production system involving other departments. To implement this kind of system successfully on a larger scale will require committed resources from IU in terms of commercial software, server hardware, programming, administration and training.

Some of the components developed for this project may continue to be usable, at least in the short term. In particular, the HTML web pages and forms themselves might be used as a starting point for the production system. Some of the *Perl* CGI code might provide a base for the glue to connect a scalable database architecture to the Web application. Finally, the *Paradox* databases and scripts might be useful for various purposes within *RUGS*, at least while they continue to use *Paradox* internally.

The difficulties Pam experienced in using the prototype system at the department level, and the ease with which *RUGS* used it, give insights into what worked and what didn't in the prototype. Here are my three main recommendations regarding implementing a second-generation, web-based, production system capable of serving multiple departments.

1. The administration of the Web-database interface is too complex and time-consuming to expect each department to be able to set up and administer themselves. Not every department can afford to set up a Web server and adapt the current system to their environment, or hire a programmer to develop a custom web forms-database interface for them. *RUGS* should bring administration of this component into their domain and make the data available back to departments electronically for their own uses. To this end, *RUGS* may need to work with participating departments to develop special custom Web pages for collecting department-specific application data.

2. It is much easier for departments to manipulate application data with familiar tools that they are already using. If the application data is only available in a specific desktop interface, departments that are not currently using that product or cannot access the data from their local database will have little incentive to use the system, because they will be stuck entering data manually into their own system. Additionally, they might have to learn a new tool to access the data and cope with two distinct databases (see below).

3. The use of separate, loosely-coupled databases for storing application data should be avoided to the greatest extent possible. The data sharing relationship between applicant, department and graduate school that occurs with the shared paper form should be mirrored electronically. Separate databases lead to inconsistent data sets, error propagation, extra administration, data loss, and other problems. Departments are going to want to maintain their own local databases (as they should), but the goal should be to have excellent sharing and interworking between the distributed department databases and the graduate school's application database, with minimal data replication. An open, client-server database architecture would be better suited to handling this type of system, for the reasons I outline below.

5.3.1 The RUGS Web/Database Server

The graduate school is planning to purchase and deploy a server for, among other uses, serving web documents. This could provide a platform for hosting the electronic application system and moving administration for the system from the CS department to *RUGS*. It also provides an opportunity to change the system from a loosely coupled set of desktop databases to the type of client-server system described above. Below are my recommendations for the *RUGS* database server.

For the *RUGS* server hardware needs, I would recommend *as a minimum*, a 90Mhz *Pentium* computer with 24Mb RAM, a two-gigabyte SCSI disk subsystem (the larger the better) and a DAT backup unit. This configuration would provide a server capable of providing Web and database service as well as file and print services, if needed, for a moderately busy server in a PC network.

Unix to *DOS* connectivity has traditionally been poor, and *DOS*'s notorious instability makes it a poor choice for a Web server. The *Windows NT* operating system fits the bill as the platform for a combined Web/database server in a PC environment. It has excellent stability and provides robust multitasking and networking. It supports both Netware (IPX) and TCP/IP out of the box and is easier to set up and maintain for non-Unix shops like *RUGS*. With Windows NT Server as an operating system, the server should be able to integrate smoothly into *RUGS*'s existing Novell network.

There are at least two freely available HTTP server programs written for Windows NT: one from *CERN*, the birthplace of the World Wide Web, and one from *EMWAC*, Microsoft's european research lab. There are also commercial Web servers available for Windows NT from companies such as *Netscape Communications*, which offer advanced features like encryption. The HTTP service would serve documents and handle CGI scripts. *Perl* is also available for NT, and many

people use it for CGI scripts on that platform, just like under *Unix*.

The database used for the *RUGS* server should be a client-server product and not a desktop database, which typically has shortcomings for the kind of scalability, multi-user access, sharing and multiple-front-end desktop connectivity needed in this application. There are a number of vendors of this kind of product, including *Oracle, Sybase, Informix, Ingres* and *Microsoft*. I do not know enough about the relative merits of these products to endorse one, although I would recommend a close look the SQL server products from Oracle, Sybase and Microsoft. The database chosen for the *RUGS* server should have the following characteristics:

1. It should allow multi-user, concurrent access by several departments, applicants (through the web) and administrators at *RUGS*.

2. It should connect to many different types of desktop databases for use as departmental clients; i.e. an open solution. The ODBC (Open Database Connectivity ?) and SQL standards are the two non-proprietary standards for client access.

3. The client's (e.g. *Paradox*, *Access*, *Foxpro*) should be able to manipulate tables on the server as though they were native tables and preferably, build local databases that combine tables on the *RUGS* server with their own tables. This means that, like the paper application, the information supplied by applicant, department and graduate school is consistent, and an update or correction by one party is reflected to all users of the data.

4. The database should provide security features that allow various types of access restrictions to be placed on records and fields so that each party can access only that portion of the data to which they are entitled, and in the manner entitled.

5. It would be a definite plus if the database scripting language could be used for the Web *CGI* interface. Connecting databases to the Web is becoming very popular, and we can expect to see commercial products for doing so appearing shortly.

The deployment of the *RUGS* server would change the nature of the electronic application system as shown in figure 4. The *RUGS* server would provide standardized application pages which applicants could fill out directly, and the application data should be accessed directly out of the server database through *Perl* or a database interface language. The application data can be accessed by the graduate school and departments using either a web interface or a database client like Microsoft *Access, Foxpro* or some other graphical front-end. Technically advanced departments would be free to implement their own databases/web pages/information system, provided it can interface with the *RUGS* database server. An easy way to do that would be to have web hyperlinks on departmental web servers pointing to the *RUGS* application home pages. *RUGS* could provide incentives for integrating new departments into the application system by offering to work with them to create special supplemental application pages for department-specific application information, offering advice on clients that work well with the information system, and training graduate secretaries in the use of the system.

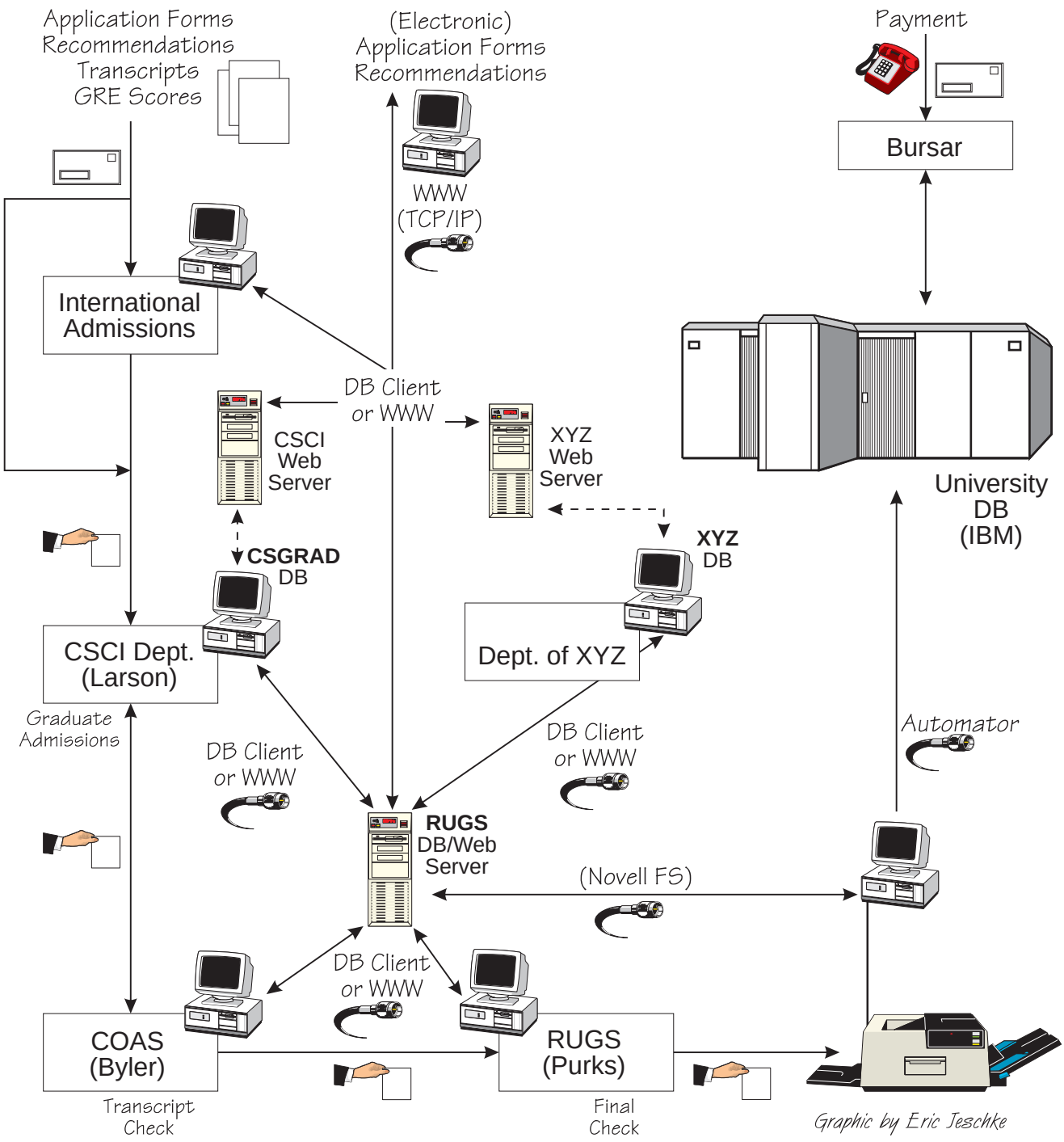

Figure 4: Second Generation Electronic Application

There are a number of important advantages to centralizing the incoming applications processing at *RUGS*. Since the graduate school has to validate the transcripts and GRE scores anyway, they may as well do it *before* the information goes out to departments. In *addition* to electronic applications, *RUGS* could receive the standard paper application forms, transcripts, and GRE scores directly. The paper applications could be scanned into the *RUGS* application database and validated against the transcripts and GRE scores; this data could then be made available to departments either electronically, on paper reports, or a some combination of both. This procedure would allow the most flexibility for both departments and applicants. For example, an applicant could apply electronically to *any* department; if the department in question is not set up to receive the data electronically from *RUGS*, *RUGS* could print out paper forms for the department with the transcripts and GRE scores already validated. Figure 5 is an example of how nicely the application data can be typeset electronically for report generation; our prototype used the LaTeXsystem to typeset reports from the application database. Other departments, like Computer Science, might choose to receive everything electronically from *RUGS*, and would benefit from not having to process the transcripts and GRE scores directly.

The office of International Admissions should be brought into the electronic application system, with access similar to other departments. Electronic admissions is especially attractive to foreign applicants, who benefit from the fast, free and simple transactions compared to international mail, which is expensive and can often take weeks to deliver. In Computer Science, the vast majority of applicants are international. E-mail inquiries we have received indicate that we would have processed quite a few more applications if the project's scope had included international admissions.

The *RUGS* GRE database should be integrated with the system and a *SPEEDE* transcript interface could be added as an interface is developed. Paper transcripts are going to be a necessary part of the system until most colleges and universities are set up to exchange them electronically. The bursar should be urged to accept payment for electronic applicants by telephone credit card payments; this could also be handled through the web interface if encryption is supported by the server.

The *Automator* mainframe interface is primitive, but effective. As long as the university databases are maintained on IBM mainframes it will continue to be useful for uploading scanned paper applications as well as providing a convenient entry point for other electronic integration.

Indiana University

Electronic Application

Signature Release Form

Name	SSN	Application No.
John Doe Jr.	999-99-9999	CSW1042

You are required to sign this form verifying the validity of the information provided in your electronic application before you will be allowed to register for classes.

Application

I certify that the information I provided in my electronic application is complete and accurate as submitted.

Letters of Recommendation

The Family Educational Rights and Privacy Act of 1974 opens many student records for the student's inspection. The law also permits students to sign a waiver relinquishing his or her rights to inspect letters of recommendation.

Below is a summary of your responses to the following question: *Do you waive your right of access to the letter of recommendation provided by the above individual?* [Y=Yes, N=No]

Recommender	Waived?
Dr. Edward Kildare	**Y**
Dr. Susan B. Anthony	**Y**
Dr. Liam Neeson	**Y**

I hereby certify that these responses are correct and in accordance with the information submitted in my electronic application.

Residency

I hereby certify that my state of legal residence as recorded in my application is **Alabama**, and that the other residency information contained therein is complete and accurate.

Signature **Date**

Figure 5: The Signature Form