This presentation will describe the functional requirements needed to send and receive student academic transcripts using the SPEEDE/EXPRESS standard. A complete implementation of SPEEDE/EXPRESS involves more than just purchasing a commercial EDI product and plugging it into the translator port of your administrative computer. In fact, the translator function is only one component of a much larger system.

Copyright 1996, 1997 Robert W. Lowe, all rights reserved.
How To Implement SPEEDE/EXPRESS 
(EDI In Education)

Background

The Metropolitan Community College District, with a current enrollment of approximately 20,000 students, is a public, tax-supported two-year college system serving Kansas City and 12 suburban Missouri school districts.

The Metropolitan Community Colleges—Longview Community College (south), Maple Woods Community College (north), Penn Valley Community College (central) and the Blue Springs and Independence (east) campuses—currently celebrate the 80th anniversary of Kansas City’s oldest institution of higher education. First organized in 1964 under its current organizational structure, the present community colleges inherit the proud tradition of the Junior College of Kansas City which was founded in 1915.

Each year The Metropolitan Community Colleges prepares approximately 25,000 paper transcripts to send to students and outside organizations such as universities, colleges and potential employers. In addition, around 20,000 transcripts are received for use in advising and for Financial Aid processing. The Metropolitan Community Colleges had considered the implementation of SPEEDE/EXPRESS during the early 1990’s. In the absence of other area colleges’ and universities’ support for EDI (Electronic Data Interchange), MCC never made it a priority to develop EDI-based software. In 1995 we were approached by Central Missouri State University and by The University of Missouri System to participate in a new pilot to begin exchanging records electronically. Although we were involved in a new student information system development project at the time, we didn’t want lose this opportunity to begin using EDI to exchange student records. It was felt that this technology would save many thousands of dollars while we continued to use our legacy system. It was also decided that by developing a way to extract student transcripts using the SPEEDE standard we would have an ideal way to populate the database of our new client server system.

Introduction To EDI

There are essentially two functions in an EDI application. These are Translation and Transmission. Everything else is just details. Translation is the process of converting information between its current form and another form. For outbound transactions, it is the process of converting the data elements from the institutional database (structure and content) into the standard EDI format. The student transcript, for example, would be converted from its Student Information System format into the American National Standards Institute (ANSI) X12 TS130 transaction set. The figure on the next page depicts a student transcript in X12 format. Of course, if a school maintained their transcripts in EDI form, translation would not be necessary. Upon receiving a request for a transcript, the Student Information System would simply transfer the records to the receiving partner. For a variety of reasons (too numerous to discuss in this paper), transcripts are not stored in this format.

For inbound transactions, translation involves conversion of the data from its EDI format to a usable form. This may be as simple as formatting and printing the document on paper. At the other extreme, it may involve conversion of the data to include course equivalency from the sending to the receiving agency so that it can be stored on the institutional database.

The transmission function involves the process of sending EDI documents to a receiving institution or receiving documents from a sender. This may be accomplished in a number of ways. For high volume local exchanges, sending a diskette or tape across town by courier would be cost effective. Similarly, the US Mail or independent parcel service could be used for more distant trading partners. However, as the volume of trading partners increases, this method of delivery is longer feasible. Can you imagine having to create several diskettes containing anywhere from one to dozens of student transcripts on a daily basis? The cost of personnel, postage and supplies may far exceed the benefits of exchanging records electronically.
The next progression would be the point-to-point electronic transmission using a modem and the public telephone system. Again, this would probably not be feasible unless the process could be automated. Without automation, you would find yourself manually connecting to each partner to exchange data. In fact, a new business service was created due to the complexities associated with this alternative. This service was known as the VAN or Value Added Network. Essentially the VAN provider acted like a central clearing house receiving transmissions from a client with the promise to deliver appropriate documents to each trading partner. With a service such as this, complexity is reduced, as long as everybody is using a VAN. You simply assemble outbound EDI transaction sets into functional groups (similar transactions and destinations), place them into an electronic envelope (containing addressing information) and transmit them to your VAN. For inbound transmissions, you would connect (via the public telephone or direct connect leased lines) to the VAN and retrieve your data from your in-box.

To accommodate this mode of exchange, the EDI standards define data segments used solely for the redistribution function. The X12 standard, for example, provides the GS/GE and the ISA/IEA segments to describe the functional groups and interchange information necessary for redirecting the EDI data to the in-box of each trading partner (or to another VAN who would provide that function). The figure below depicts how a transmission might look.
Nowadays, the Internet provides another resource to trading partners. Many schools across the nation provide Internet services to faculty and students. With little or no extra costs, schools can use their Internet connection for administrative use, including EDI. In fact many VANS provide Internet connections to their clients. Now you have the option of communicating directly with a partner or using a VAN as before. Recently, a new Internet service evolved that may greatly accelerate the use of EDI in Education. This service, provided by The University of Texas, Austin is a sort of Internet VAN. Like other VANS, the UT Austin EDI Server, as it is known, provides for the re-distribution of EDI transactions between its registered users. Anyone with MIME-capable e-mail or Internet File Transfer (FTP) capabilities can use this service. Besides the flexibility and convenience of this service, perhaps the most notable characteristic of the UT Austin service is that it’s FREE.

Of the two basic functions required to implement EDI, I feel the most complex (or costly when using the commercial VAN) is the transmission function. Now, with the advent of the University of Texas service, we are only faced with providing the translation function. The remainder of this paper will therefore focus on the details of implementing the translation function. There are several vendors who market EDI translation packages. Many commercial student information systems vendors are beginning to develop their own translator software, or are designing interfaces to the more popular commercial translator packages. You might be fortunate enough to own one of these (very rare) integrated systems. If not, you will learn that the complete implementation of SPEEDE/EXPRESS (now known as EDI in Education) involves more than purchasing a commercial EDI translator and plugging it into the back of your administrative system. In fact it is highly unlikely that your administrative system will interface tightly with any commercial translator package.

Instead, you will likely have to exchange data in batch mode between your student information system and the translator software. This is typically done using one or more flat-files that you have to create (for outbound data) or process (for inbound data). You will have to write several programs, create data structures and develop processes to accommodate this exchange. There appears to be a widespread misunderstanding about the amount of work involved in developing these interfaces. In one meeting at our colleges, I even overheard someone at a very high level say “…the Coordinating Board is going to buy EDI for all colleges in the state.” I have seen
numerous colleges who are maintaining Commercial EDI “shelfware.” They have found out the hard way that there just is no commercial turnkey solution to EDI. For this reason, some schools have chosen to develop their own translator routines completely integrated with their administrative system. For the record, I am not opposed to using commercial translator packages. In the right environment, they can be useful tools in implementing EDI in Education.

Translation Functional Requirements

During FY96, The Metropolitan Community Colleges were involved in a pilot project to develop software to provide for the exchange of student transcripts using the SPEEDE/EXPRESS (EDI) standard. Other members of the pilot project were: The University of Missouri System, Central Missouri State University and St. Louis University. We developed several programs during this project which used several data sets (files). This software included some rudimentary EDI conversion programs and transcript print programs. In addition, a suite of online based programs provided for the storing and viewing of inbound electronic transcripts and for the scheduling of outbound transcripts. Additional batch software provided for the extraction of transcripts from the MCC database for each request scheduled by the on-line system. Even though we developed our own EDI translator routines, most of these programs would still be necessary if we had used a commercial translator. The remainder of this paper will use the screen images and reports produced by our system to depict some of the requirements of the EDI translation function. Hopefully, you will gain a better understanding of the components that you may need to develop for your implementation. In addition, under certain conditions all of these programs may be licensed to other educational institutions or agencies free of charge.

The Partner File

Some of our programs need information pertaining to the school or agency where we are sending an EDI transaction. Look at the segments below. The ISA segment among other things contains addressing information for the sender and receiver. This addressing information is used by the Texas EDI Server (and VANS) to route our EDI transactions to the proper destination. We store the codes ZZ*KCMETROCC00000 (sender) and ZZ*178396 (receiver) in a file we call the Partner file. Later in figure, within the ST*130 transaction set, the N1, N2, N3, N4, and PER segments again contain information retrieved from our partner file.

ISA*00*00*ZZ*KCMETROCC00000*ZZ*178396
GS*ED*178129*960906*1017*U*00304*609060001*X*003041ED0020!
.ST*130

N1*AS*METROPOLITAN COMMUNITY COLLEGES ADM*71*178129!
N2*IN SYSTEM!
N3*3200 BROADWAY!
N4*KANSAS CITY*MO*64111!
PER*RG**TE*8167591000!
N1*AT*UNIVERSITY OF MISSOURI-COLUMBIA*71*178396!
N3*105 JESSE HALL!
N4*COLUMBIA*MO*65211!
PER*RG**TE*3148822121!

We maintain our partner file as an indexed file but it could just as well have been defined as a relational table. The random access key to the file is simply a unique 5-digit number. Other options initially considered for the key were IPEDS, FICE and ACT codes.
In addition to the addressing information shown above, other attributes are stored in the partner file. This information is depicted on the two maintenance screen-images shown on the previous page. One attribute pertains to the EDI trading status of the entity. We populated our partner master file from the crosswalk table supplied (for a fee) from the AACRAO office. This table containing thousands of potential trading
partners contains the various electronic codes for each school. Not all schools in this file are using EDI. These records are marked (by default) as Inactive in our partner file. Once we establish a trading relationship with the partner, the record is marked as Active by the on-line program. For schools who are registered with the UT Austin Server, we mark the record with a code of T. We may eventually only trade with those registered with the UT Austin server.

Sending transcripts to the Texas server is currently a manual process. That is, we have to use an Internet FTP client to connect to the server, then use FTP commands to put the file on the server. Likewise for other testing we have done, we have manually connected and transferred transcripts. Someday this process may be more automated. When that day comes, the Internet connection information stored in the partner file (depicted on the screen) may be useful. There are other attributes in the file. One indicates whether the trading partner accepts TS146 (electronic request for transcript) from our system. Another specifies whether this partner is a post-secondary school or K12 school district. These values are (or will be) used by some of the programs. For instance, we do not presently send transcripts to other than post-secondary schools. Our software rejects requests to send to K12 partners.

The EDI Prepare Queue File

A second file that we defined is known as the EDIPQUE or “Queue” file. This file again could have been defined as a relational table. We currently maintain it as an indexed file. This file contains a single record for each outbound transaction. For instance, if an operator schedules a transcript (using a on-line program described later) to be sent to a school, a single record is stored in the EDIPQUE file. The record contains all of the information necessary to join with and access the institutional database and the partner file to create a TS130 transaction set. Throughout the day or week, records which represent outbound transcripts are stored in this file by on-line programs and by Interactive Voice Response (touch-tone telephone). These queued transcripts are awaiting the batch extract process. This batch extract process is performed by program we developed called EDIMAIN.

Program EDIMAIN

The purpose of EDIMAIN is to prepare an outbound file for transmission to one or more trading partners. In its current release, the EDIMAIN program will produce transcripts in EDI format. Each record is an image of an EDI segment complete with element and segment separators. (Another version, available for licensing, will produce flat-file records for each EDI segment with fixed length data elements.) The outbound file is organized to facilitate sending to the UT Austin EDI Server for redistribution. It contains an ISA/IEA envelope for each interchange target. An interchange target is not necessarily a trading partner. For instance, an interchange target for the University of Missouri central office may receive transcripts for the University of Missouri, Kansas City, Rolla, St. Louis and Columbia campuses.

ISA…Interchange Target 1
GS .. ST*130..GE..GS..ST*131..SE..GE segments
IEA
ISA…Interchange Target 2
GS .. ST*130..GE..GS..ST*131..SE..GE segments
IEA

EDIMAIN produces this output file by sequentially reading through the EDIPQUE data set. The primary access index to the “QFILE” consists of the concatenation of the interchange receiver qualifier-code followed by the record transaction type, followed by the student identification. All of the transaction records therefore are already stored in the proper sequence necessary to produce the given output from EDIMAIN. No pre-sorting of the file is necessary. Transaction records are stored in the EDIPQUE with all transactions for a given interchange receiver stored together. Within the receiver, all transactions of the same type (0=transcript, 1=transcript acknowledgment) are stored together. The remainder of the primary key is the 30-character student identification. For example:
For each TS130 record in the EDIPQUE file, the EDIMAIN program extracts the student record from the MCC database. It performs routines to create flat-file records corresponding to the EDI segments that are needed to represent the transcript. Additional routines are then performed that convert these flat-file records to the EDI X12 standard. This two-phase approach was designed to allow other institutions who want to use this system to interface with a commercial EDI translator. After creating the TS130 segments for the student, the record is erased from the EDIPQUE file. For each type (code=1) record in the EDIPQUE file the program creates the TS131 segments for the student. The TS131 type records are stored in the EDIPQUE file by the EDILOAD program described later.

**The Audit File**

The EDI in education (SPEEDE/EXPRESS) standard provides a means to track the transmission and receipt of electronic transcripts. This acknowledgment function provides a measure of security to detect the transmission of bogus transcripts. The implementation methodology defined by the SPEEDE/EXPRESS committee called for a functional acknowledgment transaction set (TS131). Functional acknowledgment provides for a more positive acknowledgment than the X12 TS997 transaction set used in other applications. When using the functional acknowledgment, a TS131 transaction should be returned to the sender for each TS130 received. Besides matching up the control number and student identification, the TS131 contains a count of certain segments received in the EDI transcript. In effect, the functional acknowledgment provides an audit function. To
implement this audit function, some means of matching (or detecting unmatched) acknowledgments received from the trading partner is necessary. The audit file is our means of doing this. This file is maintained by two programs, EDIMAIN and EDILOAD.

During the process of producing outbound transcripts, EDIMAIN will create a record in the Audit file for each transcript produced. This record remains suspended awaiting a functional acknowledgment from the trading partner. When the acknowledgment is received, information contained in the transmission is compared (by EDILOAD program) to fields stored in the audit record. Unmatched audit records indicate that some failure has occurred. Perhaps the trading partner is not sending acknowledgments. Possibly, either the transcript or the acknowledgment was lost in transit. Whatever the problem, the use of the audit file and programs that maintain them provide a means of detecting problems.

A natural by-product of this audit function is an historical record of transcripts sent for students to various institutions. This history can enable the development of follow-up instruments by the institutional research department. Eventually we may develop reports or queries using this historical information.

The Transcript Repository File

A complete EDI system such as the one we developed during our pilot project will include various functions for handling incoming transcripts. Most community colleges, like us, have an open-door policy and do not use transcripts for admissions selection. Our primary use of secondary school transcripts is for financial aid record keeping and for student advising. Transcripts for transfer students are used for program or degree audit purposes as well as financial aid and advising functions. Universities, on the other hand, use the high-school and transfer transcripts for admittance evaluation in addition to advising and degree audit.

One function that is usually provided is the capability to print transcripts from inbound EDI transmissions. Hopefully this print function will only be necessary in isolated situations. One of the primary goals of EDI is to eliminate the human and material cost in preparing paper transcripts. A better alternative to printing a transcript is to view it on-line. This viewing function should be made available to the staff in admissions, financial aid, advising, registrar, and other departments. By storing the transcript for on-line access, it is no longer essential to have a paper copy in the student’s file.

Of course, some way to receive inbound transmissions and store them in a transcript repository will be necessary. These transcripts will come from a variety of sources, such as secondary and post-secondary schools. They will use different coding systems for their course catalog, and in some cases different grading systems (even within a single institution). Storing the transcripts in their native EDI form is one way to maintain a sender-independent transcript repository. We created such a repository. It is called the Transcript file. Once again, we used an indexed file instead of a relational table.

The file contains a single variable length record for each incoming transcript. The fixed section of the file contains the access key, search elements and identifying elements that can be used to find and select specific transcripts. This includes the student name, number, sending school name and other information. We defined alternate indexes over the student name and number fields to allow for a search-by-name and search-by-number capability. The fixed section also contains the EDI version level of the transcript. The variable section of the file contains the exact image of the student transcript segments. These segments have been strung together into a single stream of characters. The element and segment separator characters needed to parse them later is stored in the fixed section of the record by the EDILOAD program. Let’s look now at that program.

Program EDILOAD

The purpose of the EDILOAD program is to process inbound EDI transmissions. Input to the program is a data set in ANSI X12 format. The input file may contain one or more interchange sets (ISA/IEA) surrounding one or more functional group sets (GS/GE). This allows transmissions from multiple trading partners to be appended together into a single input file and processed once by the program. This is the format of the transmission...
produced by the UT Austin EDI Server. Currently EDILOAD will process transactions: TS130 Student Educational Record and TS131 Functional Acknowledgment of Student record.

For each inbound TS130, LOADEDI will insert a single record into the transcript repository described above. The primary key to the transcript repository is the concatenation of the Interchange sender identification and the transaction control number from the ST*130 header segment. This key will remain unique so long as each sending institution transmits a unique TS130 control number for their transmissions to us as called for by the standard.

In addition to storing each record in the transcript repository, EDILOAD also stores a record into the EDIPQUE data set. This record represents a transcript acknowledgment that eventually will sent be to the trading partner. This record contains information necessary to produce a functional acknowledgment (TS131). Specifically, the number of degree segments and number of course segments received in the TS130 will be saved. This information is contained in the TS131 transaction that is eventually sent back to the trading partner.

EDILOAD also processes inbound TS131 acknowledgments. If the inbound transmission contains functional acknowledgments, EDILOAD scans through the incoming segments gathering the information necessary to validate the information. For each TS131 received, program EDIMAIN (described earlier) should have stored a record in the Audit file. The EDILOAD programs searches the Audit file with a key derived from the inbound TS131. If no corresponding record is found in the Audit file, an exception message is produced. The trading partner would be notified of this exception since this error would seem to indicate that we had sent a transcript. Similarly, if an audit record is present but the control information received in the acknowledgment does not match the values in the audit record, an exception will be produced.

On-line EDI Functions

The on-line programs were designed to provide the scheduling function for outbound transcripts and the transcript viewing function for inbound transcripts. Most programs are invoked from a main menu as shown below. This menu program can be invoked from another menu. To date, not all functions shown on the menu are
complete. Additional functions are planned in the future. First, let’s look at the transcript selection and viewing features.

In the figure above, the operator has selected the Find A Transcript function. The search argument entered can be either an alphabetic or numeric value. When an alphabetic value is entered the system will build a browse list using the STUDENT-NAME index of the transcript repository. Entering a numeric value causes a browse list using the STUDENT-ID index. In the example above the browse list will be created using the STUDENT-NAME index starting with names beginning with CO. Once the browse-list is created, a viewing program is invoked. This viewing program allows the user to scroll through the list, searching for a transcript to select. To select a transcript for viewing, a selection number corresponding to the list in the left column of the screen is keyed prior to pressing the ENTER key. On the screen on the next page, the operator has selected the transcript with the item number 002. Upon selecting this item, the operator would be returned to the MENU screen to allow transcript functions to be performed on this selection.

Look back at the previous MENU screen. Assume now that the operator wants to view the selected transcript using the “MCC CUSTOM” format. The operator would simply key a value of “11” into the Command field of the menu. At this time the system would retrieve the transcript (selected earlier) from the transcript repository file. The EDI version level of this transcript would be determined from a field stored in the fixed portion of the record. Then the native EDI would be retrieved and dynamically converted into a display form. Finally a transcript viewing program would be invoked and the first screen of the transcript is displayed. The operator would then be able to view the entire transcript. The second screen on the next page is an example of the display produced by the program.
Let's move on to the outbound transcript preparation functions. These functions are designed to allow an operator to quickly and easily schedule transcripts to be delivered to a trading partner. To schedule an outbound transcript, the operator first keys the student identification number in the Value field containing a MENU Command selection of “04.” At this time, control will be passed to the SEND TRANSCRIPT program. This
program then calls the module that fetches the student record from the institutional database. The STUDENT TRANSscript (TS130) control screen shown below is then displayed. The student name and a RESTRICTIONS indicator are displayed. The operator may want to view the RESTRICTION details to determine if he should proceed. A Program Function key is used to invoke the RESTRICTIONS view program.

```
| Command: 04 Value: 555121212 |
```

```
6166-HELLO EDI WORLD
16=V123
```

```
<table>
<thead>
<tr>
<th>CICS for OS/2 Terminal V123</th>
<th>STUDENT TRANSCRIPT (TS130)</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT NUMBER</td>
<td>555121212</td>
</tr>
<tr>
<td>STUDENT NAME</td>
<td>LOWE, ROBERT, W</td>
</tr>
<tr>
<td>RESTRICTIONS</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>COLLEGE NUMBER</td>
<td>1_</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PF01-HELP</td>
<td>PF02-VIEW RESTRICTIONS</td>
</tr>
<tr>
<td>PF04-ADD SPECIAL NOTES</td>
<td>PF03-CHANGE/VIEW OPTIONS</td>
</tr>
<tr>
<td>PF07-DELETE A REQUEST</td>
<td>PF05-FIND COLLEGE</td>
</tr>
<tr>
<td>0999-HELLO WORLD</td>
<td>PF06-FETCH A REQUEST</td>
</tr>
<tr>
<td></td>
<td>PF08-OVERWRITE A REQUEST</td>
</tr>
<tr>
<td></td>
<td>PF09-RESET NOTES/OPTIONS</td>
</tr>
<tr>
<td>ID=V123</td>
<td></td>
</tr>
</tbody>
</table>
After viewing the restrictions screen if necessary, the operator could press CLEAR to return to the MENU screen and select a different student, or select a trading partner to proceed. To schedule a transcript, a number is keyed into the COLLEGE NUMBER field. This college number is a 5-digit call number which is used to retrieve the corresponding record from the PARTNER master file. In order to facilitate the scheduling of transcripts to the partners who may be selected frequently, a quick-pick list is shown on the screen with a corresponding selection number. In the example, seven colleges have been listed. For this quick-pick list, the actual 5-digit partner code is not used. Instead, a quick-pick call number is used. The program that manages this screen contains an internal table to link the short number to the actual partner call number. In the example, you can see that the operator has selected the University of Missouri from the quick pick-list.

At this point, the program retrieves the selected record from the PARTNER master file. It verifies that this partner is currently an active trading partner (from an indicator in the file). Information needed to eventually generate the EDI segments is also retrieved from the record. At this point, assuming everything is acceptable, a record representing a transcript request is inserted into the EDIPQUE file. A confirmation screen is displayed informing the operator of the status of the request. See the example on the next page. You will notice that the partner call number has replaced the quick-pick selection. This process, in effect, schedules a transcript to be delivered to that trading partner for this student. The actual extraction of the student transcript is performed by the batch process (EDIMAIN) described earlier in this paper. At any point prior to the transcript extraction process, the operator may change the request from this same screen. This is done by using the various program function keys at the bottom of the screen. By using PF7 for instance, a previously scheduled request can be deleted. The PF8 key will cause the existing request to be overwritten.

If the receiving partner is not on the quick-pick list, the operator has two options. First she could simply key in the partner call number. For instance, in our example, the partner number 01371 could have been used instead of the quick pick number “1” to select the University of Missouri. A number regularly used in this manner will eventually be remembered by the operator. Self-adhesive post-it notes are also a very handy tool for recording this kind of information. If used often enough, the partner should be added to the quick-pick-list. Another way to select a receiving partner exists in our on-line system. The operator simply keys the state code of the partner.
and presses the PF5 (FIND college) KEY. This causes the system to call a program which builds a browse list of partners for that state. Like our Find A Transcript browse described earlier in the paper, the operator can scroll through the list. To select a receiving partner, the operator would key the sequence number into the command entry field. In this example, I have selected number 59, Pittsburg State University.

<table>
<thead>
<tr>
<th>COLLEGE NUMBER:</th>
<th>0999-HELLO WORLD</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIVERSITY OF MISSOURI-COLUMBIA</td>
<td></td>
</tr>
</tbody>
</table>

and presses the PF5 (FIND college) KEY. This causes the system to call a program which builds a browse list of partners for that state. Like our Find A Transcript browse described earlier in the paper, the operator can scroll through the list. To select a receiving partner, the operator would key the sequence number into the command entry field. In this example, I have selected number 59, Pittsburg State University.
At this point, control would be returned to the Student Transcript master screen. The call number for that trading partner would be showing in the College Number field. The operator would then press enter to schedule the transcript. A single record would be inserted into the EDIPQUE file. On the next scheduled run of EDIMAIN, a transcript would be produced, and an audit record would be created. Later upon receipt of an acknowledgment, the EDILOAD program would update the audit record. The transaction would then be complete. The audit record would remain available for future reporting.

One other feature for scheduling outbound transcripts was developed by MCC. This feature provides a way for students to call our touch-tone system (known locally as MetroTouch) to schedule one or more transcripts. A quick pick option allows a student to select a receiving school without knowing the partner call number. For other partners, a list is necessary for the student to obtain the call-number. Students with restrictions are not able to use this system. This system has the potential of saving hundreds of person-hours since the transcript request is scheduled by the automated system. Eventually this same function may be developed for Web or kiosk client access.

**Current Activities**

During the early stages of developing the programs described in this paper, we determined that other colleges, universities, school districts and agencies could benefit if we were share the fruit of our efforts with them. Considerable effort went into the design of the files, screens, and programs. Organizations using any of these components would enjoy a drastic reduction in efforts by using any or all of these components. Even if an organization used a different computing platform or programming language, they could benefit from our work. Since September of 1996 MCC has been promoting this software by making available licenses to use any or all of the system at no cost. We benefit from establishing additional potential trading partners. The organizations benefit by cutting their implementation time and cost significantly.

**Training**

In addition to sharing the software and design, MCC has conducted implementation and coding seminars in the states of Kansas and Missouri. These efforts will continue as long as we can devote the resources to promoting EDI in Education. Three different seminars or workshops have been presented or coordinated. The first is an EDI Implementation Seminar. It includes an introduction to SPEEDE/EXPRESS, including benefits of using EDI. We also use this seminar to demonstrate the programs developed at MCC. A demonstration of the UT Austin server is presented. Finally, an introduction to the primary transaction sets (TS130, 131, 146, and 147) is presented. In a second workshop, participants learn the details of the TS130 transaction set. It includes a discussion of the mandatory segments, and most of the optional segments needed to encode a transcript. The participants then get to manually convert a paper transcript from their institution into EDI form using a text editor program. This EDI transcript is processed by a commercial translator package as a laboratory exercise. Two other training needs have been discussed, although as of the date of this paper, have not been conducted. The first, is a more detailed review of the software interfaces developed by MCC. This would essentially be a more in-depth review of the design of the files, programs and processes presented in this paper. The second workshop, requested by some of our trading partners in Kansas and Missouri would provide specifics on using the mapping software provided by one or more commercial vendors. Hopefully, these seminars will enable the participating schools to implement EDI In Education with much less pain than MCC endured.

**Summary**

During an EDI pilot project conducted in FY96, the Metropolitan Community Colleges have spent considerable time and effort in the implementation of EDI in Education. A complete computer system was developed that can be used in conjunction with or independently of commercial EDI translator software. This system provides for the scheduling of outbound transcripts and the storing and viewing of inbound transcripts. Both secondary and post-secondary institutions should benefit from using any or all of the system. Under certain conditions, MCC is offering a license to use the software at no charge to any educational institution or agency. With cooperative programming arrangements, and collaboration of schools, colleges, and universities, this software can hopefully be enhanced over the next few years. Organizations adopting it should save considerable costs in implementing EDI in Education.