3-1-1994

Computer-telephone integration.

Marcus W. Fath

SCHOOL OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

TECHNICAL REPORT

Number 71, March 1994

Computer-Telephone Integration

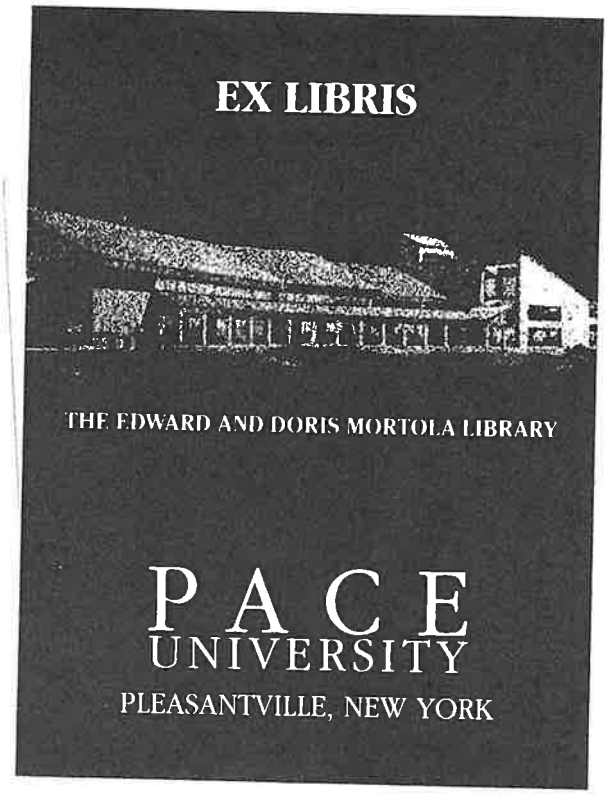
Marcus W. Fath

Department of Computer Science
Pace University
One Martine Avenue
White Plains, NY 10606

≈ and ≈

IBM Corporation
Research Triangle Park, NC

PACE UNIVERSITY



Marcus W. Fath is a graduate student in computer science at Pace University as well as an advisory programmer in the Computer Aided Telephony Systems group at IBM where he has designed and implemented CTI software since 1990. He received his B.A. degree in computer science from Rutgers University.

Computer-Telephone Integration

by

Marcus W. Fath

Background of CTI

What is CTI?

CTI stands for Computer-Telephone Integration. The merging of two technologies (voice and data) that have been, until recently, viewed as two separate entities by most company's I/S and Telecommunications departments. This merger entails connecting a company's PBX (Private Branch Exchange - a small version of the phone company's larger central office switching equipment) to their host computer in order to integrate a voice call with database information regarding the caller (commonly referred to as "screen pop"). More recently this technology has branched out into avenues such as IVRs (Interactive Voice Response Units), Predictive Dialers, and a sweet of new applications which go far beyond just popping a customer information screen when the phone rings.

A few years ago, the CTI market was promoted as one of the hottest new technologies in the United States telecommunications industry. Market research firms churned out premature reports, based more on speculation than evidence, that CTI would rapidly develop into a multi-billion dollar industry by mid-decade. Their forecasts were exceeded only by the optimism of the vendor community, which saw CTI technology "pulling through" additional multi-million dollar sales in PBX and computer platforms and related software.[1]

Over the past few years, the initial over-inflated expectations have proven unfounded. The CTI market did not grow exponentially every six months as predicted. CTI simply has been following a typical technological development path. It seems to have passed its first phase and is in the process of transitioning to a new, qualitatively different second phase- one that will absorb the initial period and expand into new areas- opening totally new vistas for products, applications, and revenues in the process.[2]

How Did CTI get started?

CTI got its start with the advent of ANI (sometimes referred to as CLID - Calling Line IDentification). ANI (Automatic Number Identification) is the process whereby the long distance telephone company provides its customers with the phone number of the person who is calling. Currently this is provided to large business customers who have direct lines connected *from* the long distance phone company *to* the business customer's PBX. These business customers are charged a small fee each time a call arrives with ANI (calls originating from analog phones or analog switches which don't contain ANI). ANI is also provided to some residential customers, for a use fee, but is limited to those customers who are connected to the same central office switch. This service has been advertised by the local phone companies as a new feature (along with call waiting, call back, etc), but requires the residential customer to purchase (or lease) special equipment to display the number.

To see how ANI works one needs to understand a little about how the telephone system works. First let's look at how ANI arrives at a customers PBX (See Figure 1).

---

[1] Voice Processing Magazine, September 1993 p. 34

[2] Voice Processing Magazine, September 1993 p. 34

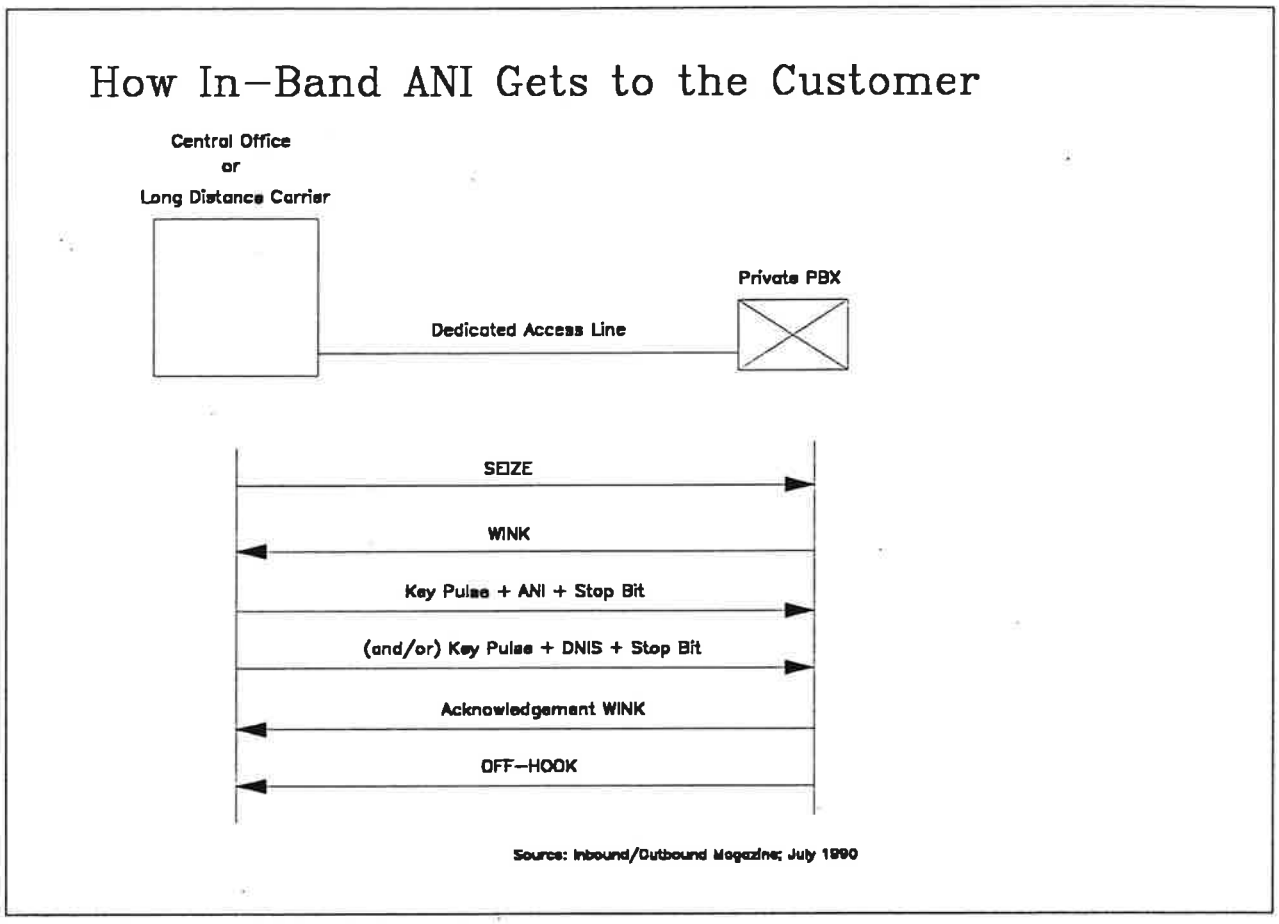


Figure 1. How In-band ANI gets to a customer.

ANI (sometimes referred to as in-band ANI) arrives from the long distance carrier via a set of very simple signals. The long distance carrier sends the customer's phone system a signal that says "Something's about to happen." That's also referred to as seizing the line. The customer's phone system responds back with a "wink" (which is more commonly referred to in telephony terminology as an ACK or acknowledgement). The long distance carrier then sends a "key pulse" (a start signal) and then the 10 digit ANI (1 + area code + phone number) plus a stop bit. When all the ANI signaling is done, the customer's phone systems acknowledges it got the digits and goes "off hook" (i.e. lifts the handset and answers the incoming call).[3]

In addition to providing ANI, the long distance carrier also provides another "phone" number called DNIS. DNIS stands for Dialed Number Identification Service. Today this number is used exclusively with the "free" 800 phone numbers and represents which 800 phone number was dialed. This is important to a PBX owner who has multiple 800 numbers and would like to know which one of them the person calling dialed so that the call can be directed to the correct person to answer it.

[3] Inbound/Outbound Magazine / June 1990 p. 6

Prior to CTI, The ACD!

Before CTI, the PBX companies of the world (AT&T, ROLM, Northern Telecom, etc) were developing sophisticated software which used DNIS to route inbound calls to the correct bank of employees who were trained to handle certain aspects of a company's business (customer service, order entry, bill inquiries, etc). Previously, a customer would call a *generic* 800 number which was answered by a person whose job was to find out why the person was calling and then transfer the call to the department which could assist that customer. This became obsolete when companies wanted to reduce costs (plus reduce the number of incorrectly transferred calls) and have these calls automatically directed to the person responsible for handling the call. An opportunity arose upon which a PBX vendor could separate his offerings from the other vendors if his PBX could somehow take the place of the 'front end' person and direct the call to the correct department.

The PBX companies determined that if a customer were to dial a *specific* 800 number, they could associate **where** the call should be transferred based on **which number** the customer dialed. The only thing PBX companies needed to do was convince each PBX owner he needed to purchase multiple 800 numbers (not a bad deal if you might have had some friendly relationship with the long distance carriers). This was not a hard sell since 800 numbers were relatively cheap; therefore the software was developed, installed in the PBX, and became commonly known as the ACD or Automatic Call Distributor. There are many different types of ACDs, some with very sophisticated routing software, but this paper will only describe the basic ACD function. To fully grasp how the basic ACD works, a few of the components that make it up need to be defined and discussed.

Most larger companies today employ quite a few people whose sole job is to answer the phone and interact in some way with the 'customer' on the other end (be it take an order for a product, listen to a customer complaint, fix an erroneous bill, etc). In the telephony industry these people are referred to as **agents**, and the place they work is referred to as a **call center**. On each agent's desk usually sits a **smart phone** (a digital phone that provides such telephony functions such as hold, mute, transfer, and conference) along with some type of computer terminal used to satisfy *why* the customer has called. These could be stand alone PCs or dumb terminals connected to a host mainframe.

The PBX itself is usually configured with multiple **ACD groups**; generally one group for each type of caller, 800 number, or department. When an external call comes into the PBX, its software determines which ACD group to route the call to based upon the DNIS received from the long distance carrier. The ACD software on the PBX then has to determine which agent in that group is available and ready for a call. This is accomplished by the agent notifying the PBX (though the use of his/her physical phone) that they are sitting at their desk *AND* they are available to take a call. In telephony terms this is achieved by **logging on to the ACD split** (sitting at their desk) and changing their work mode to **ready** (available to take the next call). Both of these actions are satisfied by dialing a predetermine sequence of numbers on the agent's phone upon which the PBX interprets and acts accordingly. Additional agent states include **logged off** (agent is not at work), **logged on after work** (agent is logged on to the ACD split but is finishing up work from the last call just handled and is not ready to take another call), and **logged on busy** (agent is logged on but not available to handle calls because they may be on a break). Once the ACD software determines which agents are **logged on** and **ready**, it then goes about determining which agent took the last call so that the one currently being routed goes to an agent who is **ready** and hasn't handled a call in the recent past.

Piecing the Basics Together

Now let's piece these many parts together into a simple (non-CTI) system (refer to Figure 2). To do this let's go through a simple scenario where a customer calls company ABC to place an order for a product. For the purpose of this example assume that company ABC has three 800 telephone numbers that are configured in the PBX as follows:

1. Service department (800-555-0001)
2. Order Entry (800-555-0002)
3. Bill Inquiries (800-555-0003)

Customer X dials 800-555-0002 to order a product from ABC. The local central office switch (to which customer X's phone is connected) passes both the ANI and DNIS on to the Signaling System 7 Transport Network (not discussed here but a network of switches and their software which get a call from the originating switch to the destination switch carrying along with it both the ANI and DNIS). The destination's local central office switch receives both the ANI and DNIS from the network and determines that the number dialed is one belonging to company ABC's PBX. The call gets passed onto the PBX with the ANI and DNIS information as previously described and illustrated in Figure 1. The PBX's ACD software takes the DNIS information and determines that the call should be transferred to ACD group number 2. The second agent in that group is available and hasn't had a call in a while so the call gets transferred to that agent's phone....which rings, the agent answers it, and helps the customer with their order. This simple non-CTI system is currently the most predominant system installed today.

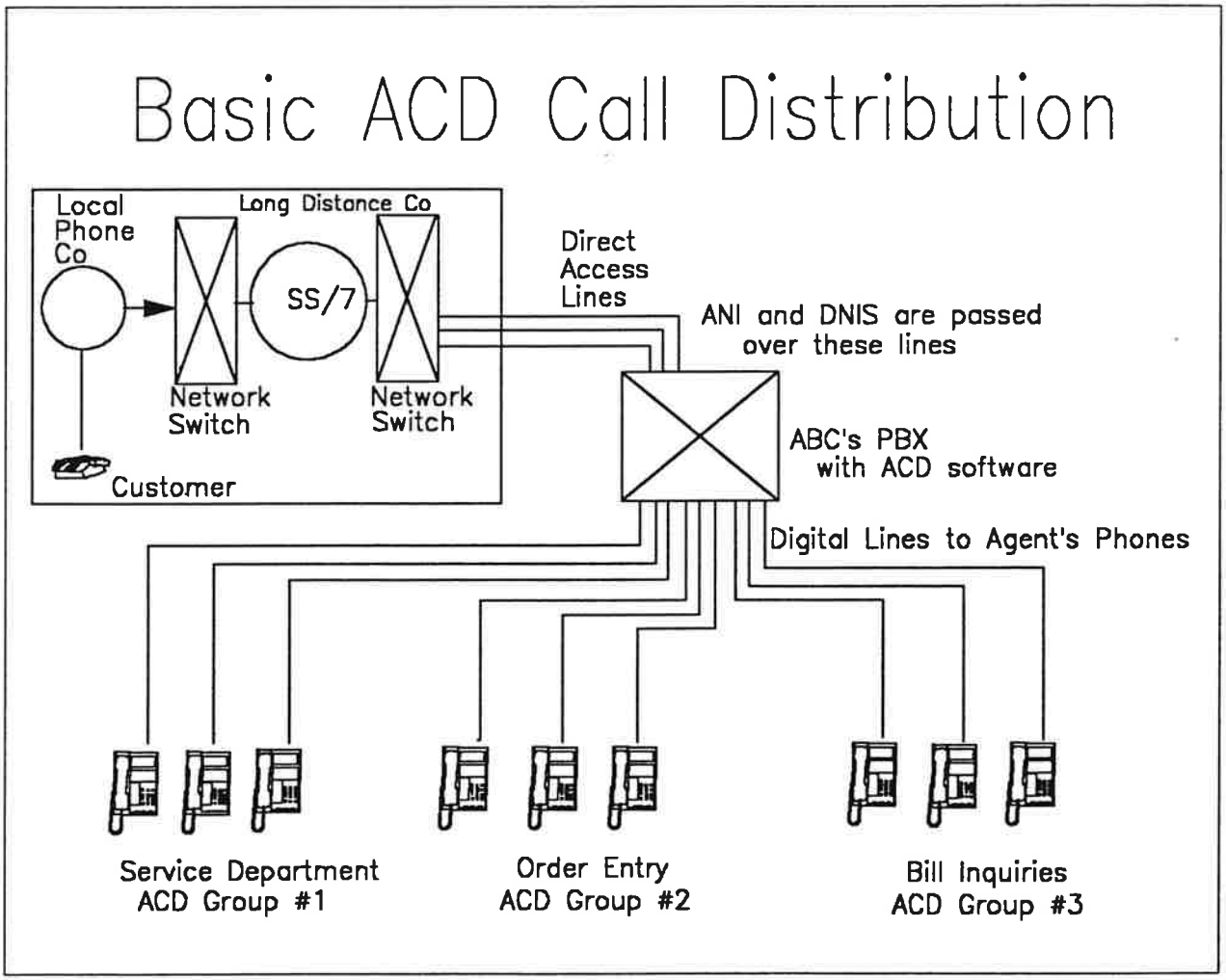


Figure 2. Basic ACD Call Distribution

The PBX Alternative, Centrex

So far this paper has concentrated on the PBX and the ability to intelligently route calls via the ACD software package. What about companies that can't afford their own PBX but would like to take advantage of the capabilities it offers? This is where Centrex steps in. Centrex is the name of the service that is offered by the local phone companies to take the place of an in-house PBX.

Let's start with a brief history of Centrex. Introduced in the 1960's, Centrex integrated a business customer's private "switchboard" with the local telephone company's central office switch. This let callers dial another office in the system without dialing the whole number. Calls could be forwarded and picked up from another phone even if users didn't share the same line. By the 1970s and early 1980s, however, AT&T and other telephone companies were moving away from the central office based Centrex. The old Bell system strategy emphasized sales of on-premises equipment, notably PBXs. Quickly Centrex capabilities lagged behind those of newer electronic PBXs. By 1984, Centrex was a mature product and on its way to decline. Upon divestiture, however, the regional Bell operating companies were prohibited from selling PBXs. This led to a renewed interest in enhancing Centrex services (to make them appear and act like on-premise PBXs).[4]

Today Centrex has several unique qualities that appeal to many small and medium size business customers. It offers the same features and capabilities that large leading-edge customers use (such as ACD call routing) while allowing the small business to acquire these capabilities with very little capital investment. It also eliminates the problems of growth since a business can expand its use of Centrex much quicker than one which owns its own PBX. One major advantage that Centrex holds today is the ability to support multiple call centers from a single central office switch. Call volumes that overload one location can be switched to another call center. This capability is driving the PBX vendors of the world to begin to offer this service.[5]

While Centrex is allowing small and medium size businesses to take advantage of the ACD functions a PBX offers, its one major drawback is that it does not provide the Adjunct to Switch link (discussed next). Inroads have been made to add these capabilities, but thus far these efforts have fallen short as the Ajunct capabilities are not widely available nor as functionally rich as their PBX siblings. Until this link can be established, the main focus will still be on PBXs (as is this paper).

---

[4] Voice Processing Magazine/May 1992 p. 42

[5] Voice Processing Magazine/May 1992 p. 43

The Next Piece, The Adjunct

The Adjunct is a computer which is physically connected to a telephone switching system (currently PBX systems and very limited central office systems) and communicates to the switch giving it switching commands. These commands instruct the system to answer phones, transfer calls, conference calls, route calls that come into the ACD etc. Adjuncts came about due to the fact that most PBX manufacturers were looking for a platform that would allow them to respond quickly to ever-changing market demands. As with most major manufacturers, the planning, development, and introduction of new products takes in excess of a year. The Adjunct provides a mechanism by which any company can customize the use of their PBX without having to rely on the PBX manufacturer incorporating the company's needs into the next set of products they produce.

The Adjunct-Switch connection is a two way relationship in that the Adjunct issues commands to the switch to control the phone operations while the switch in turn passes back event streams (which contain DNIS and ANI) indicating what is taking place with the physical phones. This is where Computer-Telephone Integration flourishes. For many years, a company's vast amount of data has been held by the I/S department. The telephone system has been controlled by the Telecommunications department. The ability to merge these two separate systems/technologies and allow them to work as one is the major focus of this paper.

First let's take a look at how a typical CTI installation is configured (see Figure 3). Coming into a company's PBX are DALs or Dedicated Access Lines. This is the physical connections (24 digital lines per T1) between the central office switch and the PBX. These private lines are not used by the long distance phone company for any other reason than to route a company's calls directly to their PBX, and for making calls from the PBX to outside the company. Connected to the PBX are both the Adjunct computer (also known as the host computer) and an IVR/VRU (Interactive Voice Response/Voice Response Unit). The IVR/VRU will be discussed later in this paper. One can also see that the agent's phone is directly connected to the PBX while their workstation is connected to the host computer. This is where CTI takes over. How can the phone system be used to assist with the data entry/retrieval of an agent? This is where the Adjunct-Switch connection becomes a vital link.

The single most unique identification item an individual owns is his/her social security number. Unfortunately the PBX doesn't provide this automatically when someone picks up the phone. What it does provide is ANI, which in the **data** community, is the next best thing to uniquely identifying an individual. If the database connected to the host computer has the customer's phone number as one of the relational fields of the customer record, then providing ANI on a database search could provide the agent with the customer information record without having to first query the customer for their identification.

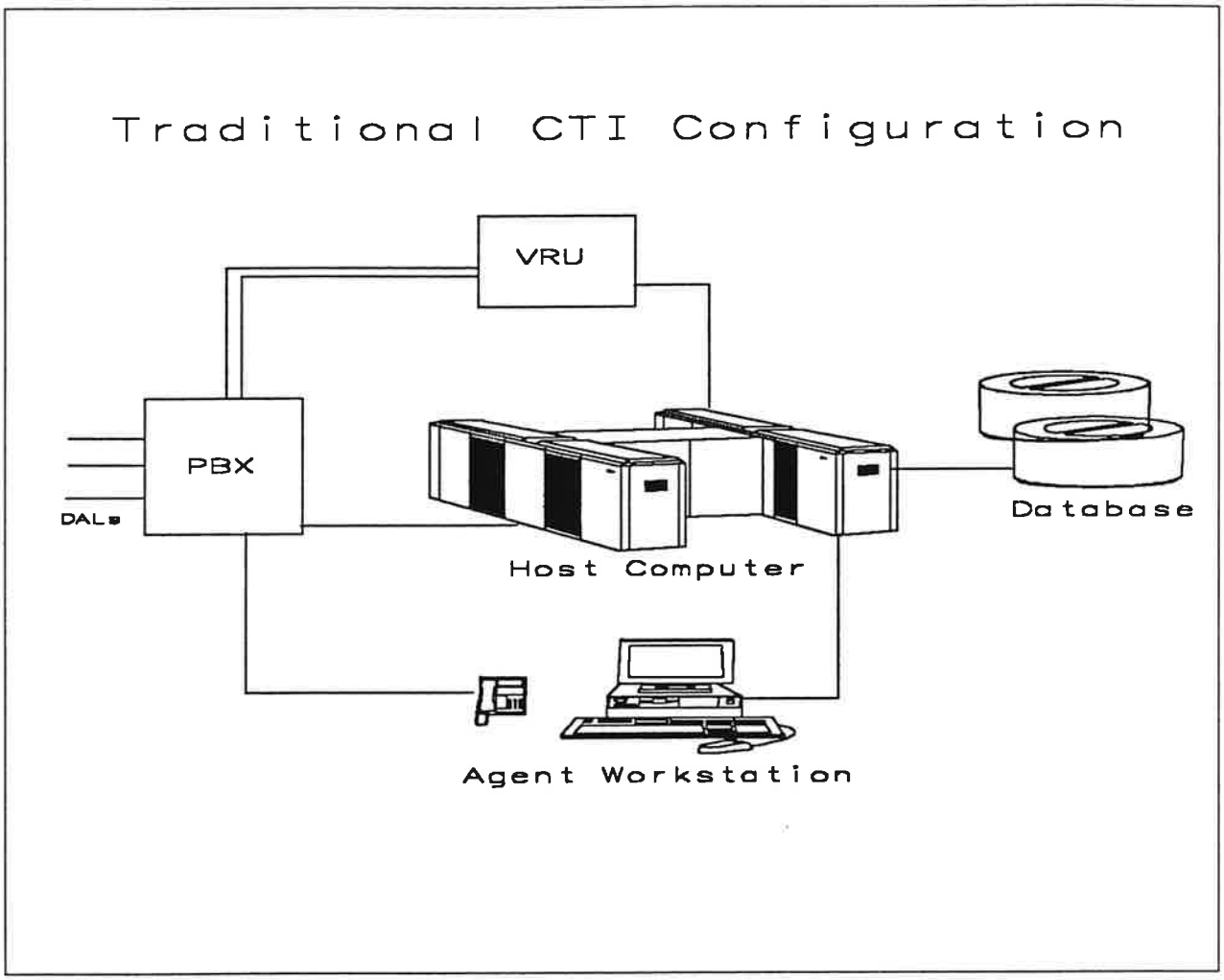


Figure 3. Basic CTI Configuration

This *lookup* scenario just described is more commonly referred to as **screen pop**. The PBX provides the host computer with the ANI at the time the call comes into the ACD. While the ACD is deciding which agent to transfer the call to, the host computer is out doing a query on its database retrieving the customer's information. The ACD then decides upon an agent to transfer the call and notifies the host computer which agent it has chosen. The host computer takes the customer information it has retrieved and sends it to the terminal of the agent whose phone has just been rung by the PBX. Usually within the same second the agent's phone rings he/she is provided with the customer's information. Previously a customer's records were retrieved during the first 20-30 seconds of the call after the agent asked the customer a question or two (what is your account number? what is your social security number? what is your phone number?), and then entered this information into their host application. Human error when entering customer input combined with the problem of the *customer* not having the required information at hand obviously lengthened the initial call setup. CTI, and the advantage of eliminating querying the customer as to his/her identity, has been found to save between 18 and 28 seconds per call. The only thing an agent must do is verify the screen pop to ensure he/she is speaking with the right person. The few seconds saved doesn't seem like a great deal of time but to call centers that average 5000 - 7000 calls an hour, a savings of 23 seconds (on the average) could result in a higher volume of call traffic, or a lower 800 phone bill. A rough calculation would show that a call center which averages 5000 calls in an 8 hour period (saving 23 seconds a call) would save 115,000 seconds (or almost 32 hours worth of long distance charges!). It isn't difficult to see the benefits and cost justification of this technology.

Unfortunately, early manufacturers and software developers weren't able or willing to envision and provide applications far beyond this generic, horizontal application. This, along with lack of market exposure, has

contributed to the slow growth of CTI and its lack of acceptance. Within the last year or two things have begun to change. Not only are businesses beginning to use and rely on this technology, many small and large companies are seeing this as one of the growing markets for the 90's. Toward the end of this paper I will highlight a few of the areas (and uses) where CTI has grown.

Using the Adjunct Intelligently

As mentioned above, the Adjunct-Switch connection is a two way pipe. Listed in Table 1 is a generic set of commands that can be issued to the switch. One should note that a majority of these commands are very similar to buttons that reside on the physical phone and replicate actions the agent may perform manually. By providing a software application which replaces manual actions, the physical phones that reside on the agent's desks will soon become obsolete. With the cost of a typical ISDN (Integrated Services Digital Network) phone in the range of $500-$900, companies have great incentives to move to a software solution rather than remain in the process of purchasing (or leasing) these digital phones.

Table 1 (Page 1 of 2). Basic Adjunct Commands to the PBX

Adjunct Command Issued to PBX	Description of Command
Answer Call	Used by an application program to answer an incoming telephone call on behalf of the agent. This enables the phone to be answered without the tradition "lifting of the handset".
Conference Call	Used by an application program to retrieve a held telephone call and join the held call with a second **active** call.
Disconnect	Used by an application program to remove a party from its current telephone call.
Extend Call (Consult)	Used by an application program to place an existing **active** phone call on hold and attempt a connection to another party (making a call to third party). Referred to as Consult because an agent may wish to place a customer on hold, consult with a supervisor, and then return to the original call.
Hold Call	Used by an application program to place an existing **active** phone call on hold.
Invoke Feature	Used by an application program to invoke a specific feature on behalf of the agent or the phone. This may include logging the agent on, logging the agent off, or changing his/her work mode. It may also include activating the Do Not Disturb feature of the phone, turning the message waiting indicator light on, or enabling the forwarding feature which forwards all calls destined for a particular agent's phone to another extension.
Make Call	Used by an application program to establish a phone call between two parties. This is used in place of picking up the handset and dialing a number.

Table 1 (Page 2 of 2). Basic Adjunct Commands to the PBX	
Adjunct Command Issued to PBX	**Description of Command**
Park Call	Used by an application program to park a call. In layman's terms, it enables the agent to put the call on hold at one physical phone, walk down the hall to another room where certain reference material may be, and pick up the same call from a phone in that room.
Pick Call	Used by an application program to pick a call. In layman's terms, it allows an agent who is not currently on a phone call to connect to an incoming phone call that is ringing at another agents desk. This is useful if an agent walks away from their desk without making themselves unavailable to take calls. Their phone will ring from a customer and they will not be at their desk to take the call. Therefore, allowing another agent to *pick* the call, this second agent does not have to leave their desk and go over to the "missing" agent's desk to answer the call. Instead they can just *redirect* the call to themselves.
Redirect Call	Used by an application program to change the destination of an incoming telephone call once it reaches the ACD software. This is useful if a company wants to write its own "ACD" software routing scheme for routing inbound calls.
Retrieve Call	Used by an application program to retrieve a phone call that has been placed on hold. This can be used to reconnect to a call that had been *extended*. (See Extend Call above)
Transfer Call	Used by an application program to retrieve a held telephone call and transfer that party to a second **active** call. The agent doing the transfer does not remain in the call (remaining in the call would be a conference call request).

Table 2 contains a sampling of the events that come from the switch which notify an application program of when the call comes into the switch, and also what actions are currently happening with the call. These are usually reported by the switch either after a request has been processed from the application (requests from Table 1) and that action is successful, or if the agent **manually** executes one of the commands at their physical phone. By reporting what is taking place during manual interactions, the switch is allowing an application program to follow the progress of a call even if an agent chooses not to use the software solution provided to execute the phone functions.

Table 2. Basic PBX Events received by the Adjunct	
PBX Event Provided to Adjunct	**Description of Event**
Call Alerting Event	Indicates to the program that a telephone call has been assigned to an agent, and that agent's telephone is *ringing*.
Call Conference Event	Indicates to the program that two telephone calls have been joined, or conferenced together.
Call Connected Event	Indicates to the program that an agent has become an active participant in a phone call.
Call Held Event	Indicates to the program that a call has been placed on hold by one of the parties in the call.
Call Parked Event	Indicates to the program that a call has been parked by one of the parties in the call.
Call Picked Event	Indicates to the program that a call has been picked by a second agent. This event usually follows an Alerting Event indicating the first agent's phone was *ringing*.
Call Routed Event	Indicates to the program that a call has been routed to a particular party. This event is usually seen after the call comes into the ACD software and the ACD software *routes the call to the next available agent.*
Call Transferred Event	Indicates to the program that a call has been transferred from one party to another party.
Disconnected Event	Indicates to the program that a party's connection in a particular phone call has been terminated.
Feature Invoked Event	Indicates to the program that a particular feature has been invoked at an agent's phone. See Invoke Feature in table 1 for a description of the types of **features** that can be invoked.
Network Reached Event	Indicates to the program that an outgoing call has progressed to the point where the local switch has connected to the long distance company's switch or the Central Office switch.
Call Setup Event	Indicates to the program that a telephone has been lifted "off hook".

In addition to the requests an agent may issue and the events that get reported as the result of these requests (or manual interaction), the adjunct is also provided with a mechanism to query the switch for various pieces of information. Table 3 outlines some of the requests an application may issue to obtain information regarding the status or configuration of the switch.

Table 3. Basic Query Capabilities of PBX by the Adjunct	
Adjunct Query	**Description**
Query ACD Information	Used by an application program to interrogate various aspects of the ACD such as: 1. How many agents are logged into the ACD. 2. Which agents are logged into the ACD. 3. How many calls are currently waiting to be serviced by the ACD.
Query Hardware Status	Used by an application program to interrogate various aspects of the physical switch environment such as: 1. How many trunks (direct access lines) are currently configured. 2. What is the time of day clock set to on the switch.
Query Party Status	Used by an application program to solicit information about a particular agent such as: 1. How many calls is this agent currently involved in. 2. What state is the agent in any of the above calls (actively participating, have a person on hold, etc).

Adding the next CTI piece, The IVR/VRU

Imagine how people would react if every business they had to deal with gave them a user friendly computer instead of a person. Many would turn and run but the majority are growing accustomed to it. People today are living in a world where 24 hour access to their vital information is becoming crucial. That is what interactive voice response does. But instead of giving each customer a computer, they use a touch tone phone to access and update most database information.[6]

Banks use IVR/VRU to give their customers 24-hour access to bank statements or make credit card and loan payments; airlines use it to provide flight information and schedule changes; cable companies use it for billing inquiries, pay-per-view ordering and installation verification. Nearly every service-oriented business with information to offer could use IVR/VRU technology.[7]

The greatest advantage of IVR/VRUs is that they handle the routine data inquiries without the need to talk to a live agent. Referring back to Figure 3, you can see that the IVR/VRU is connected to both the PBX and the host computer. Instead of routing a particular 800 number to an agent, the PBX may route it first to the IVR/VRU. Once there the IVR/VRU 'speaks' to the person calling stepping it through menus of actions it can accomplish (look up an account balance, order a movie etc.) Almost everyone today has dealt

[6] Call Center Magazine / September 1992 p. 32

[7] Call Center Magazine / September 1992 p. 32

with this automated agent at one time or another. The beauty of the IVR/VRU is that is can query most data that resides on the host while also transferring the call back to the PBX (and on to a live agent) if the user requests it. This technology pays for itself simply because the more mundane tasks agents handle can be off loaded to the IVR/VRU. This results in less agents being needed to handle as much, if not more, traffic than exists today.

The technology behind the IVR/VRU is somewhat simple. Most IVR/VRUs are simple computers which have multiple analog telephone lines connected to them. In the simplest of terms they are a very sophisticated set of telephones! Running on these computers is specialized software which retrieves from the hard drive compressed voice segments, links them together, and plays them as sentences to the user listening on the phone. The IVR/VRU is also *listening* for the user to "talk back" by hitting the touch tone numbers on the phone, or in some more sophisticated IVR/VRUs, using speach recognition technology to take the spoken numberic and translate it into a touchtone key. The major difference between IVR/VRUs are the voice segments that get played and the actions that result from the user hitting his/her touch tone keys. Most IVR/VRU solutions end up becoming customized solutions because of the specialized voice segments that get played (eg. "Welcome to Nations Bank"), and the access method and form of the way data is stored (on the mainframe, LAN server, etc) and presented.

Also built into the IVR/VRU software are some of the more basic types of commands found in Table 1. The two most commonly used commands are the Disconnet (to hang up when the transaction is complete), and the Transfer (when the customer has chosen to talk to a live agent). The Transfer command usually transfers the call back to a predetermined ACD number on the PBX which in turn determines the next available agent to pass the call on to.

The beauty of the IVR/VRU is that a call center can now handle a much larger call volume without having to hire additional agents. The IVR/VRU doesn't call in sick nor does it need benefits or want a different job. It is also much cheaper to increase the capacity of an IVR/VRU when a business grows than add additional agents.

Current CTI Uses/Application Examples

I would like to give you brief examples of how CTI is being used today in the market place. Some of these examples are generically described because the company which has actually implemented the solution believes they have a strategic advantage which they do not want to share with their competitors.

1. Dallas Texas has created the **Job Machine**. It is an IVR/VRU which lists hundreds of current job openings, at no cost to the caller. A job seeker calls the job line to not only hear job listings, but also apply for the job right on the spot. The system makes the touch tone phone a computer keyboard and the caller is prompted to type in relevant information to create a personal file.[8]

2. An IVR/VRU system from Paragon might just put Charles Schwab out of business. The IVR/VRU system allows customers to dial in to buy/sell stock, seek out options, get quotes, assemble a portfolio, and get e-mail messages from the broker. This system allows the typical telephone caller to do what the PC user has been able to do for years.[9]

3. A major student loan company has a 1 - 2 punch for customers who forget to repay their loans. The first punch comes from an automated outbound dialing program which places calls on behalf of agents and only connects the agents to calls which are answered. This has increased the the efficiency of the agent as they are freed from having to dial the phone and don't have to wait for someone 'not to answer' the phone. The second punch comes when a customer calls in requesting information about their loan. If the ANI matches one of a customer who is late in paying, the call is transferred immediately to agents who can handle these cases and council the customer in how they can more effectively pay their loan.

4. A major credit card company uses ANI to pop a customized customer information screen of the person who is calling depending on which 800 number the customer dialed. If the customer dialed the credit card company's 800 number which is for billing information, then the customer's account information is presented to the agent when he/she answers the phone.

5. Most banks today have some form of IVR/VRU which allows customers to call up and inquire about account balances, last "x" number of deposits, what checks have cleared etc.

6. Several catalog ordering companies use ANI to route callers to the same agents every time a customer calls. This allows them to establish more of a personal relationship with the customer. The screen pop may provide the customer's last 5 purchases (in addition to other more pertinent information) which the agent can inquire whether the customer was satisfied with the products he/she purchased. A customer may also need to be routed to a particular agent because he/she only speaks a foreign language and needs the attention of a particular agent.

7. A North East power company uses IVR/VRU technology to assist with power outages. If a customer calls the power company to notify them that the power is out, the IVR/VRU software kicks in and determines if future callers are from the same area. If they are, it plays a prerecorded announcement that the company is already aware of the problem and provides the customer with an estimated time of repair. This system frees the power company's agents up so that if 5000 people are affected in one area and another 5000 are affected in a second area minutes later, the power company's agents aren't busy answering questions about the first problem and never being notified of the second.

And the list goes on and on....

---

[8] Voice Processing Magazine/ January 1993 p.15

[9] Voice Processing Magazine/January 1993 p. 16

Example of a Recent Customer Installation

Finally I would like to illustrate and explain a CTI installation that I recently participated in. I am unable to give the name of the customer, but I am able to describe their configuration and mode of operation. The customer we dealt with wanted to 'test the waters' to see if CTI was as good as advertised. To do this they decided to start small by automating their help desk located in Maryland. There are 12 agents who staff the help desk 24 hours a day, 7 days a week. Their primary activities include: resetting passwords, restarting links between machines, scheduling system engineers to fix hardware problems, and answering a variety of questions regarding I/S services.

The first application they requested was an intelligent screen pop which provided the help desk agent with both personel information regarding the employee calling in addition to some basic I/S information (terminal address, user id, etc) kept about the employee in their databases. The second application they wanted was an automated agent which could perform some of the more routine tasks the live agents handled (such as resetting a password).

Prior to our CTI installation, the help desk agents had *dumb* terminals on their desks which were connected directly into their 3090 mainframe. These terminals allowed them to query the status of any of their systems in addition to issuing operator commands which allowed them to perform their job's required tasks. The configuration for the new system that was proposed and implemented by IBM appears in Figure 5.

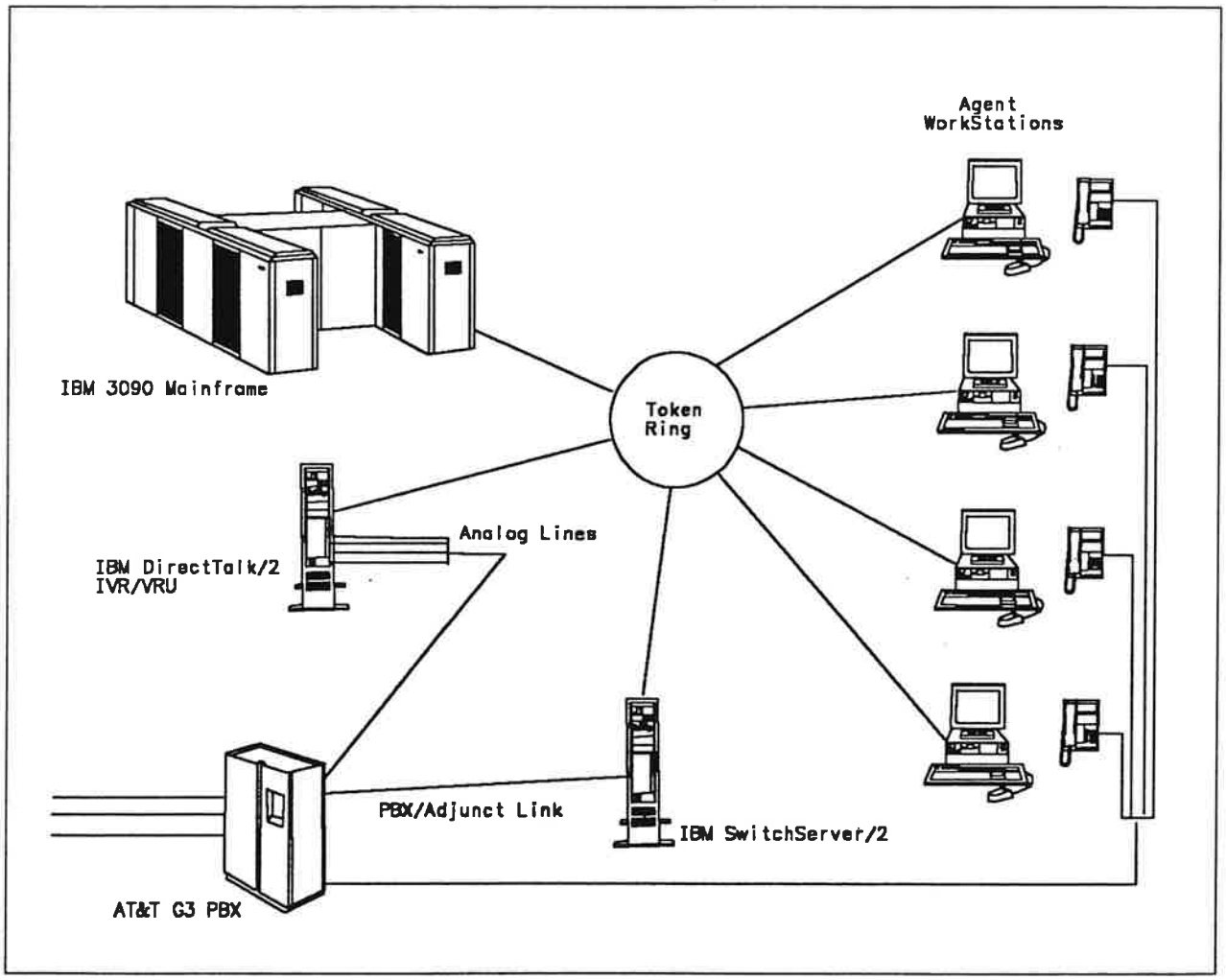


Figure 4. Current Customer Installation

Starting with the agents, we replaced their *dumb* terminals with PS/2 workstations. These were needed to run the CTI software application and communicate with our SwitchServer/2 product. The SwitchServer/2 product itself does not issue commands to the switch. Instead, it can be seen more as a protocol translator as IBM has devised its own architecture, called CallPath, which defines how the commands and events in Tables 1 and 2 should be encoded. The reasoning behind this is that it allows an application programmer to code to one "protocol" of commands and events without having to code to each PBX's protocol. A second reasoning is that SwitchServer/2 provides a consistent set of events to depict what is happening with a call. Not all PBX manufacturers provide their *events* in the same order nor with the same information content. SwitchServer/2 keeps a call state model which allows it to provide the same event flow regardless of the event order (and content) of the PBX it is connected to. Thus the workstations sitting at the agents desks have a CallPath application running on them which issues CallPath commands (such as answer the phone, put the call on hold, etc) to the SwitchServer/2 product which in turn translates them (in this case) into the AT&T format.

Also attached to the system is a DirectTalk/2 IVR/VRU. This machine became the 'front end' agent for the entire system. All calls to the help desk are first routed to the IVR/VRU. Since this customer is not paying for ANI (as mentioned before, a small charge is incurred for every call received with ANI), the IVR/VRU first prompts the caller to enter his/her employee number. Once this is verified on the mainframe (through a database lookup), the customer is prompted with a menu of actions that can be taken. From these actions the customer can choose to reset their own password, reset a hung terminal, or be transferred to a help desk agent for any other problem. If the last option is selected, the IVR/VRU transfers the call back to an ACD number on the PBX which chooses the next available agent to give the call to. Something to note though, there is also an application running on the SwitchServer/2 machine which is talking to the IVR/VRU. This application collects the customer information record retrieved by the IVR/VRU and passes it on to the proper workstation once the call gets transferred.

Now I'll walk through a few of the typical scenarios that take place with this help desk:

Customer calls the help desk to reset his/her password.

1. PBX directs call to the IVR/VRU.
2. IVR/VRU plays script which prompts customer to enter their employee number on their touch tone phone.
3. IVR/VRU passes this number (over the token ring) to an database lookup application running on the mainframe.
4. IVR/VRU receives back from the mainframe application a customer record. IVR/VRU prompts customer to verify the contents of the record.
5. If verification is successful, IVR/VRU prompts again to ensure customer wants password reset.
6. Upon receive of 'yes' from customer, IVR/VRU sends password reset command off to mainframe application.
7. IVR/VRU then instructs customer to see their manager to receive their new password and disconnects the call.

Customer calls the help desk to reset his/her terminal port.

1. Exact same steps as above but instead IVR/VRU sends command to mainframe to reset the customers terminal port.

Customer has question about a software package.

1. Identical first four steps as in first scenario.
2. IVR/VRU transfers call back to PBX.
3. ACD software on PBX sees a call come into it. PBX sends a *routed* event to SwitchServer/2 stating that a call has arrived at this ACD group. SwitchServer/2 translates and passes this onto the appli-

cation. Application sends a command over to the IVR/VRU to retrieve the customer information record retrieved for this call.

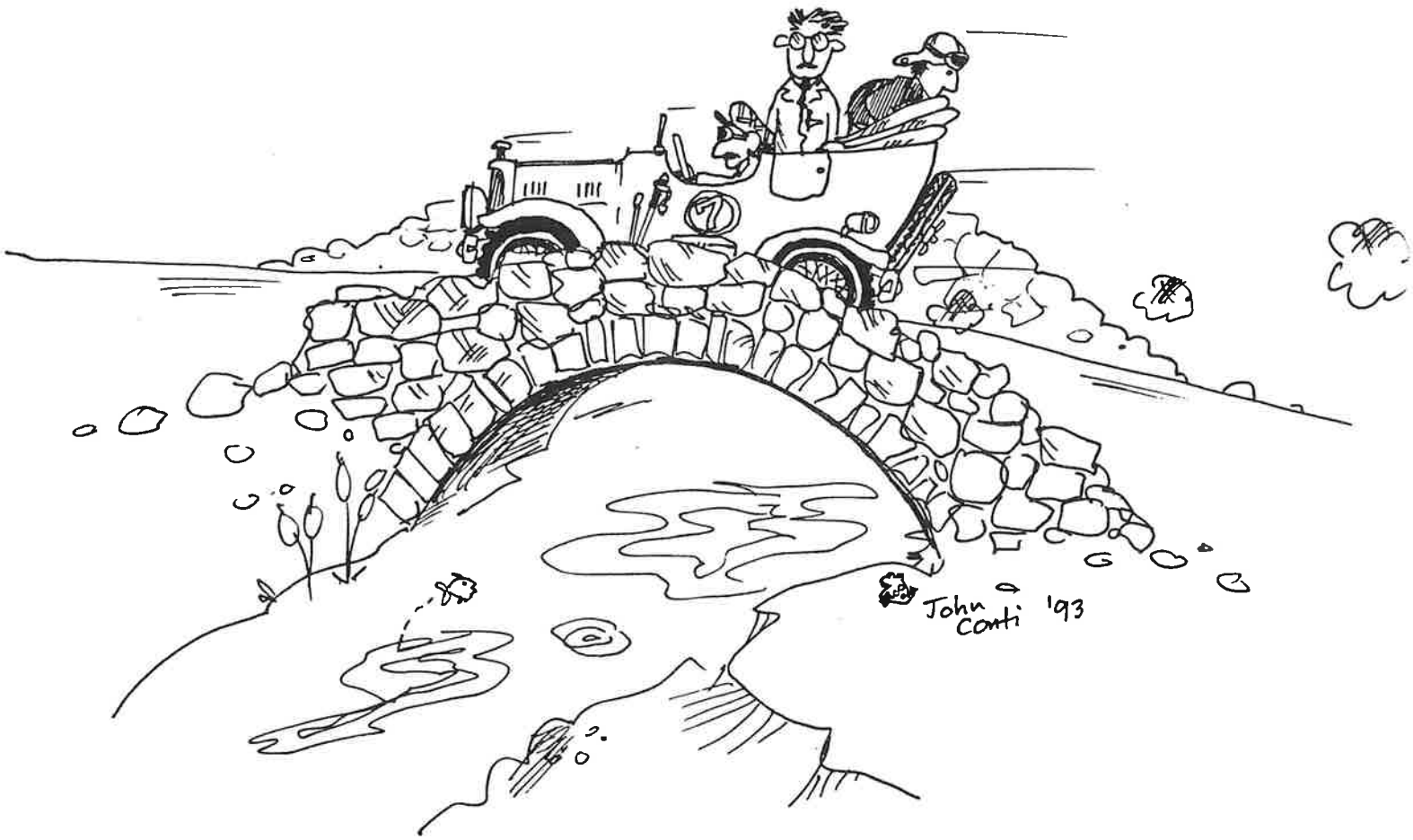
4. ACD software selects an agent and transfers the call to that agent. At the same time, PBX notifies SwitchServer/2 where the call is being transferred to. SwitchServer/2 translates the *alerting* event, sends it to the application, which in turn sends the customer information to the terminal sitting next to the phone that is ringing.
5. Application on agent workstation receives customer information and displays it to agent (**screen pop**).
6. Agent at terminal uses their mouse to select the answer call option (button) on the graphical user interface application that is running on their workstation.
7. *Answer Call* request comes over to SwitchServer/2 (via Token Ring) which translates it and forwards it on to the switch.
8. Switch responds by taking the phone off hook (most agents wear headsets so they don't have to lift a phone) and sends a *connected* event back to SwitchServer/2 which translates and forwards it on to the application running at the agent workstation.
9. Any further agent commands are treated the same. This includes issuing a Disconnect request to hang up the phone when the conversation has ended.

Conclusion

While CTI offers great advances in integrating our purely **data** world with our telecommunications **voice** world, it still is fighting an uphill battle. Many **data** organizations within a company are hesitant in allowing a new interface accessing the data they store. It is somewhat amusing how protective these people are of *their data.* On the other hand, the telecommunications branch of many companies is reluctant to allow a somewhat new technology to take over the operation of their phone system. Current PBX systems today offer redundancy to prevent failure and are rarely out of service. Allowing this new software to run the phone system makes them uneasy if it were to crash. Most would rather wait for others to *weed the problems* before they jump into this new arena. Thus there is a great reistance to trying something new.

Where does that leave us? As with most new technologies, its acceptance will happen over time. CTI has already proven itself from a business perspective (cost savings, labor savings, etc), but it has not been established long enough to prove that it will not fail when needed the most. One of the most critical resources a company owns is its data. The ability to communicate is also a critical resource. If either one of these fails, a company may fail.

Slowly more and more companies are beginning to use this technology. They see the advantages far outweigh the costs and are willing to take some risk and work with CTI vendors to make this technology both stable and acceptable. Once one company establishes a strategic advantage using CTI, others scamble to take advantage of it or put their survival at risk. Companies already realize that to remain profitable many of their practices will have to be streamlined or changed. CTI will end up being one of their key pieces of change.



Before Alan Turing, Computability Studies were conducted using "Touring Machines"