New Fangled Phone Systems Pose New Challenges for System Administrators

Snoopy
iXOS Software GmbH

New Fangled Phone Systems Pose New Challenges for System Administrators

Snoopy – iXOS Software GmbH

ABSTRACT

Recently I have noticed a remarkable shift in the way companies deal with telephone systems. More often than not the selection, installation, programming and above all administration are put in the hands of the systems administration groups. This paper gives an introduction to the architecture and features of NFPS (new fangled phone systems) which are rather different to the usual systems we have to take care of. Additionally some of the new language you will encounter is explained. And to round it off I will try to relate some lessons learned on installation day as well as how we utilize features of the new phone system to improve our daily system administration.

Background Information and Problem Definition

iXOS Software is a smallish German software house. We currently support a user population of about 200 users in our headquarter in Munich with a sysadmin group of five. We run a large network with some 500 machines, PCs and workstations as well as numerous WAN connections in order to give our users Internet access and connectivity to our subsidiary companies in Leipzig (Germany), Switzerland, Prague (Czech Republic) and the USA (Belmont, CA, Dallas and Philadelphia).

Our old telephone system (PBX) was made by Siemens-Nixdorf, a large German manufacturer of such systems. The PBX was entirely analogue, although we could and did connect some ISDN lines in order to give our directors phones with more flexibility. We were faced with the rapid growth of our company and the old system was simply just getting full: all the available expansion slots for line interfaces were getting full.

We were not happy with the quality of the voice transmission: customers complained the quality was lousy, too low in volume, hissy voice transmission etc. Dialling became a painful exercise: as the cabinet filled up, it took the processor of the PBX longer and longer to poll all the boards and so it took several attempts at dialling a number for the PBX to notice it was being asked to dial.

Additionally we were annoyed at the financial problems arising from running the old phone system. The price of the expansion cabinet with a suitable processor upgrade ran to well over 100k $. Over the (five year) lifetime of the PBX the price for an expansion board (offering a mere five phone lines) rose from about 1,000 $ to over 2,500 $. We felt this was the wrong direction the prices were moving given the prevailing technological trends and this certainly did not increase our customer satisfaction.

So we had to either bite the bullet and invest heavily or invest heavily and bite the bullet and decide to go for an entirely new phone system. As market deregulation has begun in Germany we were confident we could find a manufacturer of PBXes who could deliver a suitable and cheaper solution.

Basic Requirements

Our new phone system was meant to provide the following features:

- Fully digital transmission of voice in house and to the outside world (via ISDN)
- New, sexy phones with a display capability
- Good integration into our network environment (i.e., we wanted the PBX to speak TCP/IP on an Ethernet)
- Geared for CTI (Computer Telephony Integration)
- Easy administration – we want to be able to do most of that ourselves
- Advanced features – like hotlines, call vectoring etc.
- Voice Mail system
- Distributed systems – we wanted the architecture to deal with the rapidly expanding company – over several buildings on different streets and sites which can be several hundred miles from each other.

So we went out and contacted some manufacturers asking for an offer. The initial selection of manufacturers was done by simply asking around: we deal with a large number of suppliers who do a lot of selling by phone and I asked them what make phone system they were using and also if they were happy with it. This gave us an initial list of seven relevant manufacturers – these were invited to present their wares and introduce their systems.

Buying a phone system is very similar to having a baby: you are suddenly confronted with the workings of a a whole new industry with its own rules and vocabulary. The experience is very similar to buying a pram: you walk into the shop with your spouse and you found a pram of suitable size, shape and styling and price tag ($499, special deal). Bliss. Then the attentive sales lady asks you: don't you think you will need an umbrella (to shield baby from the sun), yes of course. And a collapsible roof against the rain (yes, that too). And a fur sack to keep baby warm in those awful European winters. Sigh. Yes, by all means (which in this context means – your means). By the time you have completed the system configuration and scaling you have exceeded the initial budget by a factor of about 2.5 and have also purchased mission critical sub-systems you never knew existed, let alone deemed necessary.

As said: enter the realm of phone systems as a proven but mainly UNIX oriented system administrator you find yourself in a terribly similar situation – an irrevocable process (your boss) takes over your life by delivering a new, utterly unknown system (the NFPS) onto your doorstep and you have to live with it for a long time and give it your best shot.

So lets go through the features we wanted in more detail and explain some of the language along the way.

To ISDN or not ISDN?

Firstly the connection to the outside world: well we went for an ISDN line known as a PMX – primary multiplexer. I believe in the US this is known as a T1 link. Basically it provides 30 ISDN channels of 64 kbps to an aggregate bandwidth of 2 Mbps. In Germany each 64 kbps channel is known as S0 (S zero) and the T1 is known as an S2M (affectionately also simply referred to as a PMX).

We were considering for a while to equip each and every desk with a proper ISDN phone. This would have given us some nice features across the ISDN network (like for example automatic call back if the person we want to speak to is currently busy). However it would have meant rewiring the buildings with four wire phone sockets because an S0 ISDN socket needs four wires. Some manufacturers allow you to run a two wire phone and they then mimic ISDN over these wires. However the needed adapters drive up the price of a phone by about 15-20%.

Furthermore we were thinking of equipping the PC workplace of our users with ISDN cards and using the ISDN line to the desk both for telephony via the PC and data transmission. Since such PC ISDN cards are notoriously fussy when it comes to the ISDN parameters we were not confident that foreign manufacturers of the ISDN mimicry gadgets would be accurate enough to satisfy the cards.

ISDN phones turned out to be expensive beasts and the interface cards needed in the PBX were more expensive too (higher price, less lines per card). Added with the re-wiring costs (estimated to be about $70,000 for our site) plus the risk of not being a good solution for the PCs anyway we decided to abandon the ISDN to the desk idea fairly early on in the game.

Basically this coincides with the manufacturers interests: they are true traitors of the ISDN concept because making ISDN on the desk cheap would create an open system where you could cut out the phone system and leave the phones out there. This would give you a lot more flexibility to throw out a phone system you don't like. But the manufacturers make this a ridiculously expensive option and point out with coy battings of their corporate eyelids that their two-wire digital phones are vastly cheaper than the ISDN solution. But of course you are then stuck with their phones too and are locked into their proprietary digital protocols: you cannot then change manufacturer of the PBX without changing all the phones. This makes a change prohibitively expensive.

So the pressure is on: once you have made your choice of phone system you can be assured that it will be around for a lot longer than your run of the mill UNIX workstation which gets thrown out after a year or perhaps two. It is therefore vital that you buy yourself into a corporate strategy which will fit your plans for years to come. Take into account the growth of the company (iXOS grew at 50% p.a. since its founding in 1988) and you have to find a PBX which will scale well and handle the anticipated loads. Serviceability, availability of spare parts, training and administration all become very much more important. The investment into the PBX is not only significant due to the amount of money changing hands but also due to the time involved before you can ever change it: therefore it is wise to invest a lot more in the training of the admin staff etc etc. for such a system.

Incidentally: as not all public switches are digital yet in Germany, we have kept four analogue lines as a backup – and indeed our local exchange is analogue which means within our area code we can only be reached via the analogue lines from the immediate vicinity.

Basic Features of PBXes

PBXes are usually UNIX based systems – usually PECOS UNIX which is a real time UNIX. variant. Before you go overboard with the naive view that PBXes are just like any other UNIX system you can handle you have to be aware of the fact that the UNIX system is almost entirely hidden from your grubby palms. All manufacturers supply a custom built interface – screen/mask oriented. A shell is not available (and none of the manufacturers I asked knew what a shell was...)

In order to administer such a system you have to get certain passwords from the manufacturer: these are part of strictly controlled hierarchy. The passwords which are akin to the root password (i.e., you can do anything) are not available.

In most cases the customers are only allowed very little access and the service department of the manufacturer handles most change requests via remote modem access to the PBX. However most of the manufacturers I spoke to came in one of two categories: those who said its their phone system and they let you use it and they do all the admin. This is not such an enticing option as it might sound for their response time may be vastly different to what you want and your users expect. So the second category of manufacturers says: we spent a lot of time and money giving you an interface for you to handle the PBX and we expect you to do most of the administration yourself. This is a good idea (the one we preferred) for it means our level of service is also good for the phone system (which may or may not be better to what the manufacturer can offer...)

Hardware Features

On the hardware side of things the PBXes are essentially one huge backplane of some industry-standard bus system (usually VME). Boards can be inserted and taken out while the systems are running: special backplane logic and power wiring makes sure the boards do not get fried or the backplane hangs during the process of insertion or removal. The bus bandwidth is quite high because you have to shift data packets quickly across the back plane in order to properly transmit voice. Packets have to be delivered in sequence and connections have to be made quickly. Hence most boards have own processors (typically M68000's which are now cheap commodity chips).

Special boards provide an extension of the system bus over fiber-optic links to more remote expansion cabinets (known as carriers). This means the PBX can be viewed as a master carrier with sub-carriers which have no local intelligence bar the on-board processors of the interface boards. Such an architecture is very flexible: picture the scenario where the Telecom people tell you something like: we are terribly sorry, but we have no more line capacity for the feed of your building with the master carrier for the second set of ISDN lines. We could only give you lines to the building on the other end of your campus.

This is no problem as long as said building has a sub-carrier: you simply stick the ISDN interface board into the sub-carrier and the main carrier will route calls over the optical fiber to go out on the other ISDN lines provided the ones attached to the main carrier are all occupied.

PBXes are designed for 100% uptime: hence you can stick boards in while the system is in use: the PBX auto-detects the new board and away you go. Of course a failure of the CPU board or other critical components will stop your PBX dead in its tracks but this only happens rarely. Its advised to hook the PBX up to an uninterruptible power supply (UPS). This increases the lifetime of the system and also allows phone calls to continue for a few minutes if the power fails (the phones are powered by the PBX through the two wires).

More Terminology

Some more terminology to keep you happy: interface boards have ports. These are the two digital wires which lead to the terminal which is the fancy way of referring to a phone. Ports are usually but not necessarily associated with extensions – an extension is the set of digits you dial to talk to a principal which is the person sitting there waiting for the phone to ring. Most PBXes will support virtual extensions – this is an extension without a port. This is actually a lot more useful than it sounds: they can be used for employees who now work off site but still want their extension to be valid and to receive voice mail. But you can free up the port and give someone else a real phone.

Sometimes a phone/principal combination is referred to as an agent – sometimes agent is considered to be a synonym for a principal – this is often a context sensitive distinction. Often agent is referred to a somewhat lesser breed of phone user than a principal – often implying something akin to a boss/secretary relationship. Agents come in droves known as splits or hunt-groups which are extensions grouped logically together by the PBX to do fancy things (explained in more detail later).

Creeping Featurism

Modern PBXes are packed with features most of which you will hardly ever need. In fact most manufacturers nowadays refer to their systems as Call Centers. The difference between the two is essentially marketing driven.

Basic features of all examined call centers are things like call forwarding – you can just route your phone to another extension, i.e., where you will be seated for the next two hours of Solaris install.

Auto-Call Back means if an extension you call is busy, then you activate the ACB button. As soon as the desired extension puts the phone down, the PBX will ring your number and his/hers and connect you. The display reflects this saying something like ''Call back to Snoopy''.

Our stations have three so-called call appearance buttons (with associated LEDs). When a call arrives then the display gets activated and the call appearance button lights up. The third call-appearance button is labelled a restricted call-appearance. This means normal calls cannot arrive there. Consider if you have two calls on your extension and you want to transfer one or both out of your station to someone else. In order to do this you have to get a third free line (this is somewhat analogous to the last free process slot in UNIX being reserved for the super-user) to call the other station and transfer the call. Hence you do not want a normal call to arrive on the restricted appearance lest you will not be able to get rid of any calls short of putting the darn phone down.

If you are talking on one line then your station is considered to be active – if you have all lines (you may of course have as many call appearances as you wish or have buttons for) handling a call except the restricted appearance only then are you considered busy which means the PBX will not hand you an additionally arriving call but will return a busy signal to the caller.

Some stations (like us system admins, he, he, he) can place priority calls which means we can wind up on the restricted call appearance and/or punch through any call forwarding or other cloaking devices the principal has enabled. This is really useful if you have to notify a user that you have to take his/her workstation down immediately because its misbehaving on the net or whatever.

Conference Calls are also a nice feature – you can get up to six different people on the phone all at the same time. Our directors make frequent use of this to have phone meetings amongst all the directors of our subsidiaries. In Germany incidentally it is forbidden to set up conference calls off-switch which means we are not allowed effectively to call external numbers and get them in the conference. This law is an ancient relic and a source of great amusement even to the Telecom employees who cheerfully admit its an idiotic law which currently cannot be policed anyway because a conference call is not different to their switches than a normal call. Of course this law will go away with increasing deregulation in other words by 1998.

Every extension has associated with it a coverage path – this is a path a call can take if the principal does not pick up the phone. Our default here is to route calls directly into the voice mail after a number of rings (typically five rings.) Of course you can mention other extensions in the coverage path, up to three such points are allowed in a coverage path and you can also chain coverage paths together. Alas while you can decide when you want a coverage path to take hold for all calls, external calls only or when you are active or busy or whatever there is no way to have two coverage paths associated with an extension: one for internal calls and one for external calls.

A typical coverage path for sales people goes to the sales assistant (if the primary target, i.e., the extension that was dialled) does not pick up. Then if the sales assistant does not pick up the next point in the coverage path is the switchboard. Only if the switchboard is unable to pick up does the caller wind up in the primary targets voice mailbox. Germans are not well accustomed to answering machines and we find that its way better to humanly handle calls because otherwise customers just put the phone down.

Our phones used to have a button we programmed to be the send-all-to-cover button: all calls would directly drop into the coverage path (which for most of our developers meant voice mail). Some of our developers are real hermits and had the send-all enabled all the time, self stylized paranoiacs. So soon our customers started complaining, that our folks were not reachable (and some of our developers will not read e-mail for weeks or also ignore little red voicemail lights on their phones for weeks on end), So we took their send-all buttons away from them to enforce managements idea of reachability by phone. This created a bit of a upheaval but the waves of discontent soon ebbed.

Of the more razzly-dazzly features voice mail is actually terribly useful and the voice mail system provides a good interface to actually forward voice mails and file them away in complex directory structures. Opposed to normal e-mail, voice mail has an automatic expiry (set at two weeks at our site). Of course voice mail is yet another biblical plague – you now have an additional mailbox to wade through.

Hunt groups are useful too: you can designate bunches of extensions to be considered to be linked together. Consider two extensions 1122 and 1150 to be in a hunt group. Someone calls 1122 but nobody there picks up the phone. The PBX will then try to ring extension 1150: this implicit routing is transparent to the user. You can also designate a bunch of extensions as a a terminating extension: when a call comes in for any one extension of such a group all of these phones ring at the same time! This is actually fairly spooky to have about 15 phones go off at once in a large office. It creates instant panic and is the poor mans implementation of Captain Kirks red alert.

Hotlines and ACD Functions

One really good feature of modern call centers are ACD (Automatic Call Distribution) functions to support hotlines or telesales applications.

ACD distributes incoming calls evenly and fairly over a group of extensions organized in a split or hunt group. Usually agents have to login to the split in order to be recognized that they are available to be allocated calls. Incoming calls are divided fairly amongst all logged in agents. Due to this login process the PBX knows which agents are actually staffed.

An important accessory feature to ACD is after-call – this allows you to specify a time after the completion of a call until the agent is made available to take calls again. Its a sort of delayed login after a call. Imagine a scenario where your sysadmins take calls from the users and have to enter a request into a request system: the PBX takes the agent out of the split for a fixed time after the actual phone call (say three minutes) to allow one to fill out forms or the screen mask or whatever. When the after-call time has elapsed, the agent is automatically put back into the split in available mode.

Good PBXes provide detailed reports on the throughput of the ACD groups: how many calls are taken an hour, the average wait time, the longest wait time of a call, the number of failed calls (a failed call is one where the caller puts the phone down before an agent can take the call). Such statistics allow you to properly size the number of agents in a split to handle calls. Such things are not very useful for a small fry company like iXOS but its very useful for professional telesales companies for example in case of adverts on TV (''Call now to order your Super Thigh Stepper for the smashing price of only $139,99!''). Such adverts generate tens of thousands of calls within the two minutes after the advert. So its essential to have a powerful call center in place capable of handling such loads.

ACD functions are best augmented by advanced call vectoring features. Call vectoring enables you to program a path (N.B. don't confuse this with a coverage path) the call is meant to take amongst different splits of agents and to be treated differently. For example, you can program the vector to take into account queue lengths of the split thus enabling easy overflow functions. Some PBXes only allow such overflow functions by going down to the PBX and taking a screw driver and twisting a potentiometer to shift incoming calls from one split to the other if one of them is being flooded with calls. This is far too rudimentary; we really wanted to handle this type of problem in a better way by appropriate programming.

Vectors can take the queue length into account differently too, for example you can test ''if calls_waiting in split > 3 then deliver busy''. This would cause the fourth caller into a split to get a busy signal.

Other vector primitives include functions like testing if a call is external or internal, checking the time of day etc. This is useful for handling and routing calls appropriately. For example our weekly sysadmin meeting is on Tuesdays between 11:00 and 12:00 a.m. – we have therefore programmed the hotline vector to play a special announcement during this time so that users will fall into the voice mail immediately so we can cope with the problem after meeting.

Vectors can be interactive – you can play an announcement (''Please type 1 to leave a voice mail message, type 2 to speak to someone directly'') and then gather the input and branch accordingly. This can also be combined with the announcement of the queue length (''The average wait time is currently 3 minutes. If you wish to stay in the queue please press 1'')

Advanced PBXes allow you to integrate a voice recognition system into the vector programming. This is an expensive co-processor running to the cost of about $50,000 extra. However this can be very useful to speed up customer service in large call centers (airlines tend to be very fond of these things).

Vector programming is a bit of an art and a different etiquette applies. In my first attempts I actually did some very rude things like delivering the busy signal after people hung around on the phone for two minutes, paying the bill and listening to music. Well, it was a bug in my vector admittedly but the users reactions did make it very clear what I should avoid in the future!

Advanced vectoring allows you to also change the priority of a call in the queue depending on the number of the caller (this is only possible on an ISDN network where callers can choose to transmit a packet containing their ISDN number). This feature is called ANI (Automatic Number Identification). In a vector you can test for the number and give it special treatment. For example SAP (a company that holds a 10% stake in iXOS) has the number 06227-34-0. We use this number identification to place their calls to our hotlines in a privileged position because they usually need quick service.

ANI can be used in conjunction with CTI (Computer Telephony Integration). The vector allows you to specify to send a packet out on the Ethernet when a certain number has called. This packet is then received by a server which allows you to place information on the principals PC screen.

Some things we find lacking in the methods of vector programming is testing and setting sort of ''environment variables''. As an example take a company that wants to introduce a ''quiet time'' where all calls from the outside should go to the voice mail and not disrupt the quiet workings of the company in top gear. I would like to code a vector in the form of:

```
if ($quiet) and
   (daytime is 14:00 – 16:00) then
    route all calls to hunt-group 10
           /* our AUDIX hunt group */
```

Then in the main systems menu I would just set a variable quiet and things are enabled. If we deem the experiment a failure I just unset the global variable and I never have to go back and change any of the vectors. OK, I admit I am heavily influenced by the UNIX philosophy but I would really enjoy it if this philosophy would not be as well hidden by the administration interface.

Connectivity and CTI

This brings us to the issue of the PBXes connectivity with our LAN. We have an ethernet card installed and some applications running on the PC. Actually the issue of ethernet connectivity was a real problem for most manufacturers: of the seven applicants only two (Ericsson and Lucent Technologies (ex-AT&T)) provided a satisfactory solution. All the remaining suppliers either gave us a disbelieving stare (''Why on earth would you want to have an ethernet connection to your PBX!??'') or came up with horribly kludgy "solutions" – the favorite one was: stick another ISDN card into the PBX and stick an ISDN card into a PC. Then the PBX sends its packets to the PC which in turn plays ISDN to Ethernet bridge. This however implies that the reliability of my Ethernet connection is that of the PC and to us who know the full significance of such know that this as a totally unsatisfactory solution. No thanks.

Connectivity and CTI are based on two standards TSAPI and TAPI. The latter is supported primarily by Novell, the former is the Microsoft standard. Our PBX is adherent to TAPI and we are in the process of evaluating administration software which runs on Solaris 2.X as well as front ends running on Windows PCs.

The front ends for the users offer nice integration of the PBX with the PC. Voice mail and dialling are available through the PC. Additionally you can maintain a phone and address book on the PC and dial with a mouse click.

The voice mail integration is also really nice: a voice mail message appears as an icon on your screen. Moving the mouse over the icon will enlarge it to show the statistics associated with the message: date, length, sending extension or number etc. Double clicking the message icon will deliver the voice mail by letting your phone ring and playing you the message. You can then also manipulate the message (deletion, filing, forwarding etc.) through pop up menus.

We are also planning to use the interface to the PBX to connect it to our in-house databases – here we record our sales leads and customers etc. We want to be able to dial a customers number using a function key in the display. If a customer calls the sales people we want to be able to display his/her name on the phones display and also pop up the correct customer information mask on their screen. However we have not done this yet as we are still developing the software for this; this is work in progress.

In future we would also like to use the TCP/IP connection to the PBX to be able to send faxes directly from any PC or workstation. The reverse channel should also be possible. Our aim is to also hook up all of our modems to the PBX and access them via the ethernet link (i.e., using the PBX to act as a terminal concentrator).

PBX manufacturers want to integrate the phone entirely into the PC: your voice packets get carried to the PBX over the ethernet and then out onto the public network. The aim of the PBX makers is clear: they want to get more money out of the software as their fiscal base for charging premium prices for PBXes is being eroded by more and fiercer competition. So they want to slowly shift their revenue base to include more services and software.

The PBX makers offer other reasons for this being a good idea: you save space on your desk, you get a lot more comfortable access to all kinds of wonderful functions. For example it is planned to allow every user to receive faxes on their extension: the CTI software will then display the fax bitmap on the screen.

However we are opposed to this idea for several reasons.

We believe a phone should only be augmented by the PC but should stay a separate appliance which is functional as a separate device. We support this concept for different reasons mainly that of reliability and usability.

Reliability is a big problem. PCs crash, PCs break, users fiddle their setup files and their reliability and availability is usually far inferior to properly administered servers and workstations. Furthermore power problems arise: its impossible to hook up all our PCs to an UPS for example. It would be prohibitively expensive. Our current setup allows us to run the phones on an UPS. We believe our sales people and other heavy phone users would be very upset if they would not be able to use their phones if their PC is dead. That simply happens too often for comfort.

Usability is another issue and can have possibly legal repercussions. Phones have been around for over one hundred years and most people in the civilized world have no problem walking up to a phone and using it. However imagine a scenario of your office mate suddenly grovelling to the floor with a heart attack and you have to wait until your NT workstation boots until you can call help.

We believe, even if it sounds awfully conservative that phones will be around for a while yet. CTI offers fantastic possibilities to increase your performance using the phone but it will probably never replace the phone on the desk entirely.

Administration

The administration of modern PBXes is usually done through a channel PBXes know how to handle best: phone lines. You are supplied with two modems: one for you and one for the service department of the manufacturer. This is indeed very handy and we are quite happy to let them adjust critical parameters of our PBX (like tuning timing parameters for the ISDN cards and other things at a very deep technical level).

Administration is done via a terminal emulator over the phone line and you then talk to a screen based administration tool which hides the underlying phone system entirely. Soon we want to use a Solaris server on our network to run a Solaris based admin tool which talks to the PBX over the ethernet. We hope that we can do all the administration of our switch, the AUDIX Voice Mail and additional functions like the call charge metering through the ethernet. Also we need additional scripting functions so we can specify configuration commands like "remove the send-all-calls button on all stations of type 8410 and replace it with a date-time button.'': when we enforced management policy to take away the send-all-calls button we actually had to do this by hand which is a mindblowingly tedious way to spend a morning. Once you have done that you begin to realize that things like *sed*(1) are very useful tools indeed!

Security

PBXes allow the system administrator to create privileged ports – these are allowed to access privileged functions of the PBX. The idea is that you can do some switch administration from home without a modem just by using a touch tone phone. PBXes allow most functions to be accessed not only through programmable buttons on the phone but also though key sequences referred to as feature access-codes(FAC). For example you can issue a call redirect (call forwarding) directive by typing: *04342433#. This will forward all calls (*04 is the FAC) of extension 342 to extension 433. The # is the end delimiter.

From home you would then call the privileged port (a virtual extension). You then have to validate yourself with a password (a number of three to eight digits length). You then get a line to the PBX and the world is effectively your oyster. Small wonder then that such ports are frequently the target of crackers who specialize in breaking into PBXes. Apparently in the US this is big business to defraud companies this way. The crooks get hold of the number of the privileged port(s) and then rent an apartment and ask the local Baby Bell to put in lots of phones. Then they accost customers in the street to the tune of ''Phone as much as you like for only 20 bucks'' – they then ring the privileged port of the hacked PBX and get an outside line and the company running the PBX pays the bill. This can go on for several weeks before anyone notices.

For this reason the system allows you to create such ports only with a limited lifetime – you can specify ''number of logins'' or a lifetime in days. In case of repeated login failures the port is automatically disabled and a message gets logged to the system logs.

Of course other security issues arise: your PBX is on the Internet if your companies LAN has a connection there and you have to ensure that nobody can break into your LAN and use the phone system. People could route Internet packets through your PBX and fax out or just use the phone and you pay the bill. Not a good idea.

On top of that your very own employees can defraud you much easier. They could for example call-forward their phone to their aunt in Australia and then they go home and call their extension from home. You pay the bill to Australia. Not good.

Also users can forward their voice mail off-switch so that voice mails get delivered to your answering machine at home. This can also incur heavy costs if they do this to another continent. If the voice mail system they forward their voice mail to is also configured to forward the voice mail (perhaps back to the switch it came from) you can create costly mail loops and overflowing voice mail disks on both sides of the Big Mud Puddle.

Anybody who knows your extension number can listen into your voice mail: you can protect it with a password (a number). This is a good idea. We had one incident of a rather delicate nature where a user walked up to someone's phone to actually get at their own voice mail but by mistake typed the '##' short cut to access the (unprotected) voice mail. Promptly he got a message (intended for the actual owner of the extension) which was by his girlfriend who steamily described various activities she would undertake with him if he could only make it home from the office on time. After that incident (of some embarrassment and plenty of laughter too admittedly) we quickly revised our companies security policy to embrace the voice mail system and the PBX.

Its also not a good idea to allow users to forward other peoples phones, make priority calls or override send-all-to-cover functions etc. The PBX offers two concepts for you to specify what phones are allowed to do.

The Class of Restriction (COR) governs the type of external calls principals are allowed to make. Some companies do not allow a secretary to make long distance calls. Directors are usually allowed to do this. This kind of setup has led to the introduction of a bridged call-appearance – the secretary has a button to get at the line of the boss in order to make the long distance call and hand the connection back to the boss.

We often found this in our PBX – a lot of extra features have crept in to augment or work around a restrictive feature instead of leaving both (the restriction and the work-arounds) out altogether. However at iXOS we are a fairly easy going place – everyone is allowed to make long distance and international calls for example so we disable most of the things the manufacturers sell you as a goodie. More restrictive companies however need these functions. In fact we also do restrict some of our phones to only make internal and emergency calls as soon as we had caught attendees of our training courses making hour long calls to the States from our training rooms.

The Class of Service (COS) governs what a phone is allowed to do. Most importantly it adjusts things like: is the phone allowed to make calls in the first place (or just receive them), if it is allowed to access the Feature access Codes and if it allowed to do special things like having console permissions. The console in this case refers to a special phone known as the attendant console – this is colloquially referred to as the switchboard. This console has special permissions defined in its COS. It can issue call redirects for all other extensions. It can override any send-all-calls or call forwards the users may have enabled. It is allowed to re-record announcements and there is a plethora of other functions available only at the console.

Some other terminals like our sysadmin phones (AT&T CallMasters – the second to ugliest telephones in the universe after the Veeblezork Mark III found in use on Disconnectus 5) we can also have a COS giving them console permissions, so we can get through to people and act for and on behalf of the switchboard in cases of need.

Unfortunately the manufacturer of our PBX ignored some programming principles (again). As far as I am concerned I want to be able to define COS as a special class of its own and I define which functions (and function buttons) are allowed to be contained in this. Unfortunately our switch has an implicit idea of what buttons certain devices are and are not allowed to have. The console for example may not have a call-forward button. This is highly annoying because at times we want someone else to just handle incoming calls for a while.

I had to spend hours working around this problem by designating backup-consoles as a split and defining an ACD queue which only ever has one member – either the console proper or one of the backup consoles. How much easier my life would have been if I could have just defined a new console COS and designated several stations to be of that COS. In the definition of said COS I would have then allowed a call-forward button and life would have been much simpler. My workaround took ages of consulting with the manufacturer and the implementation frittered away several precious function buttons on the backup consoles.

Installation Issues

This brings us to the subject of installing the PBX – this is not merely buying the thing and plugging it in. You have to have a very good idea of the initial programming and special features you want to have implemented. You should have a good idea of the functions you want to make accessible to your users through the function keys.

In this respect we found it was a key move of utter strategic brilliance (actually it was some coincidence too) to take in the sysadmin training of the manufacturer before the installation. This way we knew about hoopy features and caveats and most importantly we spoke the language of the installing technicians. This helped a great deal and on a purely social level we got to be really friendly with them because they were happy to have a customer in the know who took the whole thing seriously.

We grossly underestimated the time (and consequently money) it took to get our phone lines sorted out by our electricians. This is required because digital lines are treated more carefully – the PBX measures the lines' impedance and quality. If the criteria are not met, the PBX will simply disable the port. So, you must have the copper wires in shape before the PBX arrives. If the switch decides to disable 40 or 60% of your lines on the installation day then you will find out what life is like with a big problem on your hands. The monetary cost of tracing our lines, checking them out and preliminary installation (the fiber optic cables, the distribution panels and UPS wiring etc. etc.) and clean ups amounted to half of the entire hardware cost of the PBX including the phones (over $60,000!).

Count on ordering at least 50 more phones and 20% more ports in the beginning. We thought we were at least good at mundane activities like counting things up but in the end we found we had forgotten a number of phones and also suddenly we were confronted with requests to finally put phones in storage rooms, empty offices, shacks in the swamps etc. which never had a phone before. Think hard: it might be good idea to buy that expansion sub-carrier now with your initial order when you can haggle for a massive discount.

Make a good plan of when to cut over from the old switch to the new one. Tell everyone the phones will be down – we did the cutover on a Friday afternoon when things go quiet. Think hard about which phones have to come back up quickest: the switchboard, the fax machines, the secretaries and the directors. Be sure to put the sysadmin phones online last. When you are just swapping the 134th of 260 phones you don't want to be flooded with calls of the type of "my phone is funny", "my phone doesn't work", etc. No thanks.

It took us less than ten minutes to get the switchboard back up. Fax machines and secretaries were back in business within forty-five minutes, the directors phones were on-line thirty minutes later.

Distribute the user documentation along with the new phones. Not an iota later. We also put up a web page with the most important features and the documentation of our configuration.

How are you going to get rid of the old switch and phones? We got the manufacturer of the new switch to take it all back (actually they sell the things off to Third World countries). When you have to collect more than 250 phones its a good idea to have a bunch of quarters ready and -ahem- borrow some shopping trolleys from the local supermarket. These are very handy to cart around large quantities of new and old phones.

Don't forget that an installation of this type is actually also fun! OK, it was a lot of work (eight people were busy from 3 till 11 p.m. plus all the work up front). But its only rarely that you can do something so radical and novel like giving everyone a new phone and dragging them kicking and screaming into a new digital age. We all loved and the team spirit was really good. We had a blast.

Sysadmin Support by the PBX – our Hot Line

We also use the hotline/ACD function of the PBX to help us with our sysadmin duties. We have defined a sysadmin hotline number extension (in our case 911 – and its available on a function key on every phone. (The German equivalent of 911 is a different number so there is no confusion, however 911 is a well enough known concept to be easily remembered).

One of our group is always logged into the split (usually two of us so that we can sometimes leave the desk to answer calls of nature, user requests etc.). Users call the hotline using their hot key and we take their calls. The users do not have to poll our individual extensions to get a response. Via the login process the phone system knows where to route the call to our 911 number.

So what happens if I am logged in and have to leave my seat and mere seconds after I have left the room my phone starts ringing because I have left the login button depressed? Our phone system has a feature called RONA (Ring On No Answer). If a logged in agent does not pick up the call within two rings the PBX will immediately route the call to another logged in extension at highest priority. Thus it will scan successively until it finds an agent willing to handle the call.

Note that this ringing and searching is done without the caller noticing: all they hear is some announcements and the music to fill the gaps. They hear no ringing tone and ultimately they don't know or care where their call winds up as long as we answer.

All of our Call Monsters as we call them are equipped with headsets (these became really popular with the new phone system: we now have about 30 headset users).

The sysadmins staffing the sysadmin hotline act as a dual help desk and dispatcher function. Basically any requests they can handle quickly (giving advice, kissing away tears, dishing out floppies and SCSI cables etc.) are handled immediately. Anything else is entered into the request database or escalated directly to the relevant sysadmins concerned. This setup has really reduced the interrupt rate of the other sysadmins immensely who can now get some real work done.

We change roles on demand: sometimes we have a morning and afternoon shift, sometimes one of us is dispatcher and help desk for one or two entire days, just as we deem fit. In order to keep our users informed about who is on duty (911 is of course always on duty from 8 a.m. till 6 p.m.) and may be interrupted we bought one of those "Times Square" LED displays which tells people about the admin on duty to correctly route user requests being delivered by Sneaker Net.

On the whole our sysadmin hotline has been a huge success for our group and our user community. Other departments (customer support and sales) have now taken to wanting (and getting) their own hotlines and queues etc. so they can do similar things and achieve similar improvements.

Conclusion

When we first bought the new PBX we were wary if we needed such a complex beast. However our experiences with this installation and the ease of maintenance and the level of support we can now give is worth that initial outlay. Our users are very much happier with the phone system and the fact they see and notice that we take care of it.

Also our implementation of our sysadmin hotline shows how even small companies with even tinier sysadmin departments can make good use of advanced features of modern call centers. This is really not overkill but hopefully the shape of things to come.

Author Information

Snoopy is the local resident court jester, author and system and network admin for iXOS Software GmbH (in roughly that order). He has been with iXOS from the beginning in 1988 – before that he spent time time developing hardware and writing software for various German companies. His e-mail address is: Snoopy@munich.ixos.de.