Free Software For Telephony

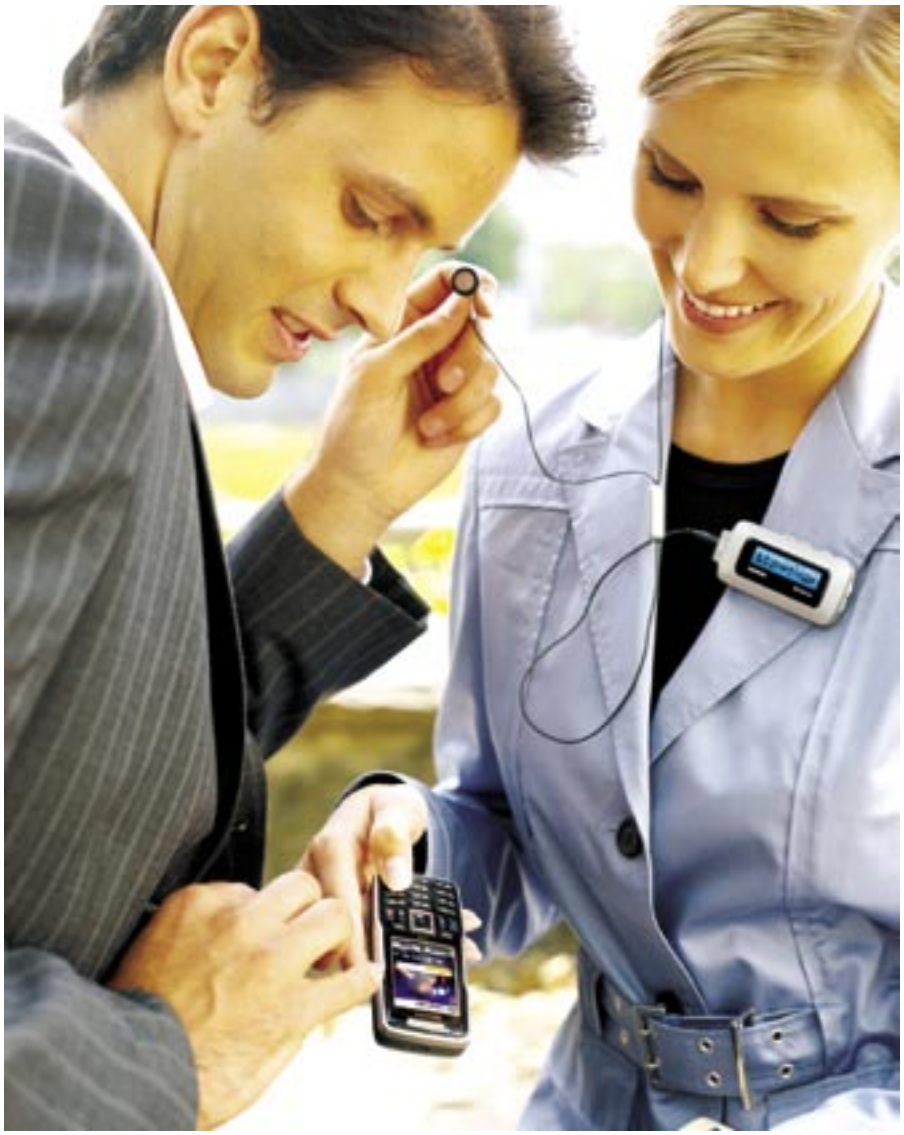
Free software for telephony has been around since the late 1960s when developers at AT&T first used a Unix application inside a PBX. However, it has taken a quarter of a century for free software to mature enough for carriers to widely accept it, especially in real-time environments such a call centres. David Mandelstam, president & CEO of Sangoma Technologies brings us up to date and explains the perspective for developers.

Broader acceptance of telephony software began in the late 1990s when the project ACS was used in combination with a Dialogic PCI card to create Interactive Voice Response applications. (The ACS later evolved to become Bayonne, a free multi-line telephony server of the GNU project.) Although the IVR software itself was free, the boards that were required to connect it to the PSTN and complete its part of the voice and signalling processing were quite expensive.

Around the same time a few other projects were started, most of them involved with the VoIP space or to be used as telephony hardware for voice, fax and modems. One of the most important of these projects was OpenH323, a project aimed at creating a full featured, interoperable, Open Source implementation of the ITU-T H.323 teleconferencing protocol that can be used by personal developers and commercial users without charge.

Adjust & Customise

A free gatekeeper module based on OpenH323 called GnuGK was created that had many advantages. For one thing, because it was based on Linux, it was cost-effective and could also be expected be reliable. The code was also adjustable and hence could be used to customise settings, coding and reactions. This flexibility resulted in many developers recognising that free telephony software could have enormous potential in commercial applications.



To DSP or not to DSP

The first indication that new technology was required for telephony applications appeared in 1996 when it became necessary to have gateways forward calls between VoIP and PSTN technologies. There were available solutions based on commercial cards, but these were very expensive since they incorporated specialised Digital Signal Processor hardware.

DSPs digitally process signals from the public telephone networks in real time. In telephony, DSPs are used for echo cancellation, translating data from one format to another or to decode the signalling protocols such as DTMF, ISDN or SS7. Their dedicated control software, known as firmware, either resides on a memory chip on the board itself or in some cases is loaded from a binary file on the host computer.

One of the major suppliers of PC-based telephony cards was Dialogic, which was recently bought by Intel. Dialogic-like cards have been around since the early 1990s during a time when the highest CPU clock rate was 50 MHz and the 8Mhz ISA bus was the universal PC standard. Having on-board DSPs to handle data locally was a must because even the smallest task, like recording or playing a file for 30 channels would completely swamp a PC of those days. The first example of telephony cards without DSPs were the so-called “soft modems”, which had marginal performance in the early days.

PC Power

In 2000, Jim Dixon of Zapata Telephony developed and built the Tormenta card, still based on the ISA bus and GNU/Linux software but without any DSP. The power of a typical processor in a PC of that time had become high enough to perform the DSP type data processing within the timing requirements of telephony networks. Using the main processor for all the data handling allowed for much lower cost cards to be used without sacrificing flexibility. Also, adjustments to software are easier and less expensive than replacing the firmware that controls a DSP.

Today there are three main vendors of

DSP-less telephony boards: Digium, Varion and Sangoma Technologies. Varion still uses one of the original Zaptel designs, while Digium has developed improved versions of the original Tormenta designs. Sangoma has designed its own high-performance boards based on its 20-year experience of data transfer through WANs. Its boards incorporate Field Programmable Gate Arrays (FPGAs) that can do some on-board processing lower to CPU loads and improve stability.

Asterisk – the first wave

In 2001, Zapata Telephony designed Tormenta 2, a more powerful PCI version of its PSTN interface card, but the company lacked the resources to manufacture it on a large scale. Linux Support Services (now Digium) took over the entire project and built it into a complete open source PBX/ IVR system called Asterisk, the first free software that put these simple, DSP-less cards to good use.

The Asterisk project has grown with the help of the open source community and outside financial investment and now supports more hardware (analogue and ISDN BRI) as well as many of VoIP protocols such as H323, SIP, IAX and MGCP. Asterisk is the most successful open source PBX/IVR system today. It supports fax, voice mail, instant messaging, transfers, conference, call queues and many other important telephony features. Under a commercial license it also supports features not available to the GPL version like patented codecs.

Asterisk is being used by developers in a wide range of applications, but most importantly to replace the classic PBX with analogue lines. Its internal design is based on the PSTN network since the signalling and the data are tightly tied. This is the reason Asterisk developed a VoIP protocol called IAX (Inter Asterisk Exchange), which is the only VoIP protocol that transports the data and signalling on the same network port.

While being mainly used on a Linux platform, Asterisk also has a Windows port through a Cygwin emulation layer for POSIX. Not yet available in wide production, it uses the VoIP protocols since drivers for Zaptel cards are not available on Microsoft's platform.

Telephony Engines to Cover all Needs

Although used successfully for enthusiasts and the SOHO market, Asterisk for commercial applications has a few issues for clients with special demands, such as telephony carriers, that have limited the technology's usefulness in larges-scale deployments.

Many features are hard-coded in and cannot be easily changed to improve performance or meet the users' demands. These users are looking for a design concept that will increase flexibility, stability and performance to support larger, more complex and more mission critical applications.

YATE

One of these "new concept" telephony agnostic projects is the Yate (Yet Another Telephony Engine) project. In Yate, telephony/VoIP protocols and other communications (like electronic mail and instant messaging) are connected to each other by passing messages through a messaging engine.

Messages carry information about the calls and other events and can be extended to cover the needs of new protocols without affecting the compatibility with the old ones.

This backwards compatibility is critical, as the whole point of leveraging free or Open Source telephony software is to reduce costs. If old protocols need to be replaced, the compelling business value of free telephony software is diminished.

Until now Windows has not been used as a platform for free telephony servers. Existing Windows-based proprietary PBX/IVR systems are still based on expensive DSP-laden cards that do most of the processing on-board. But while current Windows systems are expensive, the growth of Windows in commercial IT and communications infrastructures means the platform has to be addressed in terms of affordable telephony software.

By using Sangoma cards and its Windows drivers, Yate now provides a free, easy-to-use server for this platform that can be integrated with other software components into a powerful solution. In some cases, it is feasible to port an engine developed in Linux to the Windows platform. However, some of the engine modules have system dependencies so they have to be created from scratch for the specific system platform.

Developer Familiarity

The Yate engine is written for Windows in C++, a language that is familiar to many programmers. While developers who are used to object-oriented programming languages may need a little time to adjust, C++ offers the best compromise between balancing flexibility, optimisation and ease of writing code.

In addition, C++ hides most of the platform-specific details, allowing developers to focus on adding features without having to concern themselves with platform specifics. It is also possible to interface modules written in other languages, scripts and databases. Having the flexibility to deliver both calls and messages is important to telephony carriers who will be able to use Yate to replace other much more expensive installations.

Telephony engines are also finding their way to desktop computers, whether they are Windows or Linux. Rather than having different VoIP clients for each different protocol, the same Yate engine needs only the addition of a graphical user interface and a sound card "channel" that's only special because it supports volume control. Since it is fairly small engine, a Yate-based VoIP client can even be squeezed in the memory of a modern PDA running Windows CE or some flavour of Linux for devices, allowing a Palm Pilot or other PDA to act as a sophisticated IP telephone.

Call centres are perhaps the most demanding clients, as they need custom operator interfaces and integration with CRM applications. They can also benefit from the flexibility and extensibility of a telephony engine by passing extra data between the PBX and the operator's softphones at the time of the call. Since Yate can be built natively for both Linux and Windows, it offers optimal integration with existing applications instead of pushing a costly platform switch.

Ultimately, this will enable cost-effective commercial deployments of voice applications that can be easily integrated into existing applications that can leverage the power of VoIP in future projects.

Coming of Age

Yate and Asterisk are just milestones on a long road from expensive, closed, patented solutions to free, open and interoperable software. As the computing power of commodity hardware continues to increase, we will see increasing use of these technologies in more and more demanding applications in the future.

The work that has been done on Yate to date means that robust, free telephony software for mission-critical and commercial, real-time applications has finally come of age and is very close to becoming reality.

Founded in 1984, Sangoma develops and manufactures wide area network (WAN) communication hardware and software products, with an emphasis on the PC platform. The communications solutions, voice, data and video gateways support all popular WAN networks, line protocols and all standard PC operating systems and platforms including Linux, Windows, FreeBSD, Sun Solaris, NetWare and DOS.