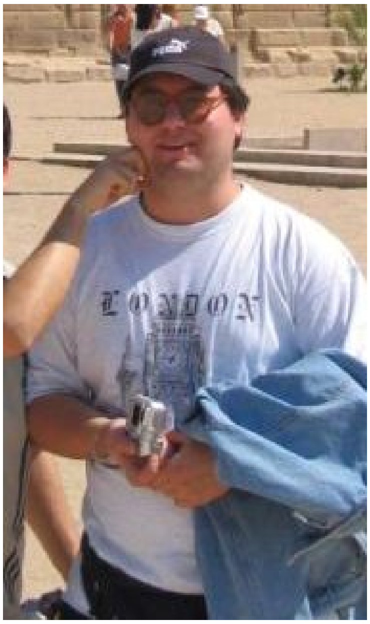
LinuxFocus article number 372
http://linuxfocus.org

Bayonne: the opensource project inside phone and IVR services



by Luca Bariani
<lucabariani/at/ferrara.linux.it>

About the author:

I was born in Ferrara (Italy) in 1977, I graduated in Ferrara in 2002 with an electronical and engineering degree. I'm a member of Ferrara Linux User Group and GNU Bayonne's group. I develop IVR and Sms phone services. Inside GNU Bayonne's project I'm writing some documentation and articles, I talk about Bayonne in some opensource meetings and manage the italian website bayonne.it.

Abstract:

Bayonne is the IVR and phone server of the GNU project, therefore a free and opensource software.

IVR stands for Interactive Voice Response. An IVR system interfaces the computer world with phone networks (with proper hardware) and can interact with commands received through them (tone signals by keys pressed or, sometimes, true voice commands). IVR systems allow for ythe creation and management of completely automated phone services with no human operator.

IVR systems and services

As of today, everyone has made a phone call to an IVR system at least once (maybe without being aware of that). In fact they are used by every telephone company (mobile or not) for their own call centres or service centres. Just think how many times you have listened to automatic and recorded voices for choosing from a menu some service, or before being switched to a human operator). IVR servers can interact in different ways with local and remote resources, and also access filesystems, databases, networks and so on... so IVR systems can be easily integrated into pre−existing systems or be part of distributed systems.

With IVR systems a lot of phone services can be created and used, some examples follow:

- Answering service: a basic phone service, it is usually implemented with simpler and cheaper systems than IVR ones, nevertheless using an IVR system a lot of possibilities and customizations become available:
    - ♦ we can manage different voice boxes for different users;
    - ♦ we can manage a remote user authentication;
    - ♦ we can add a notification (with email or anything else) for every received message;
    - ♦ we can transfer the voice messages to a remote server (through internet and/or email).
- On−line updates on highly dynamic information: IVR systems can handle dynamic informations which change day by day or hour by hour. This allows creation of information phone services, e.g. stock price list, the latest news, weather information and forecast, daily special offers for supermarkets and shops....
- On−line catalogue browsing: IVR systems can easily interface to Databases and other data sources (local or remote), so we can easily create catalogue services (e.g. for books, films, CDs, magazines and so on). Selecting category, cost and whatever else we need to, users can get lists and information about all selected items.
- Booking and purchasing services: using databases and other dynamic data sources IVR systems can offer booking and purchasing for services and goods, e.g. booking a cinema ticket or a film in a rent store, the purchasing of a good a book or a DVD. A catalogue can be included (see previous point), as well as payment by credit cards or other kinds of money (see next point).
- Payment services: in every country there are several kind of phone numbers with different costs for users:
    - ♦ "normal phone numbers": calling cost depends on the phone company, caller areas and calling time (these are the numbers which usually everyone has at home or office);
    - ♦ "free phone numbers": with no calling costs for the user (these numbers are often used by phone company's call centres);
    - ♦ "pay phone numbers": the calling cost is chosen by the phone provider and, usually, is higher than for "normal phone number"; cost can depend on calling time or be a fixed amount.

  With "pay phone numbers" IVR systems can offer phone services including user's payment, so we can sell goods and informations without credit cards or electronic money transactions.
- Event notification: IVR systems are bidirectional: they can both receive and make phone calls; making automatic and programmed calls, IVR system can notify about events through phone lines (e.g. appointments, meetings, expiring times, alarms for some important cases...).

IVR systems work on phone lines using audio for input from and output to the user, allowing a very easy integration with vocal systems like TTS (Text To Speech: audio synthesizer that converts written text to a vocal speech) and ASR (Asynchronous Speech Recognition: vocal analyser that converts human speech to written text). The firsts enable IVR systems to have vocal output from written text and not from pre−recorded voices only; the latters enable IVR systems to recognize vocal commands from input voice.

With TTS and ASR systems we can have more phone IVR services:

- Vocal electronic email: our email server receives a normal message, TTS converts it to an audio file and (after proper authentication) can by listened by the user with a phone call to the IVR system.
- Advanced answering service: ASR can improve the answering service described above with speech conversion from the voice received by phone network to a text than can be easily sent via email, put on a web server or whatever else is needed.
- Phone services for deaf and/or dumb people: we can make phone network and phone services available to people who usually cannot access them because they have are handicaped:
    - ♦ dumb people can use a normal keyboard to write a text converted by a TTS system and placed on the phone network by IVR;

- deaf people can receive a voice phone call through IVR, have it converted by an ASR system to a text and displayed on a display;
- above solutions can be combined for people that are both deaf and dumb, allowing them to make and receive phone calls.

Generally speaking, an IVR system can offer on phone network every service and every piece of information offered by a web server on a usual web site. Lots of services and possibilities can therefore be made available through IVR systems. But IVR systems can offer more: phone network and channels are bidirectional, so IVR systems can have services that call users directly (web services cannot have this feature).

Bayonne

For a long time IVR systems have been a monopoly of closed, not free software solutions by some IVR board vendors. Each of them developed IVR servers for their own hardware; Bayonne is the answer of the opensource world and of the GNU Project to this situation.

Main and official Bayonne's website is www.gnu.org/software/bayonne where you can find sources and documentation (installation and administration manuals, lots of examples and some HowTos). On bayonne.sourceforge.org there are references to external resources and the latest developer snapshot. Bayonne has a public (read only) CVS access for its sources: on savannah.org the 1.2 version (the currently stable version for production applications) is available, while on sourceforge.org the 1.3/2.0 version (the developer and experimental one) can be found.

The main purpose and characteristic of Bayonne is to offer a virtual IVR platform that is independent of the used hardware. As a matter of fact Bayonne can work with several IVR board families (not all of them yet) without needing any of them. For every IVR board family and/or vendor Bayonne features a modular driver which defines a common set of instructions and a virtual environment, for general use and without any hardware references. Within this virtual environment we can develop our IVR services without being aware of the used hardware; therefore changing IVR board or porting an IVR service to another system with a different board does not require any change in the service code.

As already stated Bayonne supports several boards, also very different ones, from 2 to 120 phone lines (analogic or digital). Moreover, Bayonne can use more boards of the same family at the same time (in the new Bayonne version we can even use board of different families at the same time), so hundred of different phone lines can be handled simultaneously. Unfortunately Bayonne doesn't support usual serial/USB modems (analog, ISDN or GSM).

To give Bayonne a try, or in general for tests, we don't need any expensive IVR board. Actually we just need a sound card as Bayonne provides a driver which emulates an IVR board and involves I/O with PC's keyboard, microphone and speakers: Bayonne can be run on your desktop PC.

In order to create IVR services, Bayonne has its own scripting language, namely CcScript (an external component for version 1.2, integrated in the main package for the developing version 1.3/2.0). This language is very simple, with complete computational features and main programming structures: conditional blocks, loops, conditioned or unconditioned jumps, subroutines.... A main feature of CcScript is to have "time bounded" instructions: before an instruction is executed the language knows its maximum execution time. For operations which involve external systems like filesystems, databases and network connections there is a timeout parameter, if the execution time exceeds the timeout, the operation fails. From the server point of view this feature is very important because it allows Bayonne to offer deterministic and reliable services.

Bayonne's computational capabilities are not limited to CcScript's ones at all. Actually Bayonne can interact with high level languages like Perl and Python without any problem. By means of this direct integration Bayonne inherits all libraries and modules available for these languages: every computational operation, usual and unusual algorithm, data manipulation, connection to database or external server (web, email, LDAP, file sharing)... So Bayonne can compute everything that can be required.

Even without Perl and Python, Bayonne can have direct connections to MySql and Postgres databases and SQL query within CcScript itself. In the same way Bayonne can access filesystem and connect via network to remote resources. Furthermore, Bayonne can directly interact with voice synthesizer TTS like Flite, Theta (only English voices available) and Festival (multilingual). Bayonne cannot currently connect to speech recognition systems, but this feature is in development status.

For the most part of IVR services, Bayonne does not need any external Text To Speech systems as it has its own multilingual vocal modules (English, French, Italian, Russian) which can speak numbers (ordinals and cardinals), dates and times (these modules and audio are completely customizable). These vocal modules include human pre–recorded voices and can be used to realize most IVR phone services directly within Bayonne's framework.

Bayonne can manage inputs from phone connections and inputs from the system itself in several ways: synchronous management, where Bayonne waits for a well known input (e.g. waits for a key pressed by users), and asynchronous management, where Bayonne receives a notification when a well known event happens (e.g. a key pressed or a timeout expired) without waiting for it explicitly. These two input management modes allow Bayonne to react quickly to user and system events without any waste of time.

As already stated Bayonne has been developed to manage a lot of phone lines at the same time, so it features commands and instructions for multithreading management. Every thread can interact with others by means of signals. This allows thread synchronization and data exchange between threads and, furthermore, connecting each other two threads which use different phone lines (so e.g. two users can talk to each other through Bayonne).

All these features allow Bayonne to be useful in a wide range of situations. It can be used for little IVR phone services with few phone lines, as well as for big service centres handling hundreds of lines. With Bayonne we can easily have "fast and simples" services or complex services involving also external systems. So Bayonne is the right IVR server for every need of IVR services.

Webpages maintained by the LinuxFocus Editor team © Luca Bariani "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org	Translation information: en ––> –– : Luca Bariani <lucabariani/at/ferrara.linux.it>

2005–04–02, generated by lfparser_pdf version 2.51