PROCEEDINGS OF SCIENCE

e-control: new concepts for remote control of VLBI-telescopes and first experiences at Wettzell

A. Neidhardt, M. Ettl
Forschungseinrichtung Satellitengeodäsie, Technische Universität München,
Geodätisches Observatorium Wettzell
Sackenrieder Str. 25, D-93444 Bad Kötzting, Germany

C. Plötz*, M. Mühlbauer, R. Dassing, H. Hase
Bundesamt für Kartographie und Geodäsie, Geodätisches Observatorium Wettzell
Sackenrieder Str. 25, D-93444 Bad Kötzting, Germany

S. Sobarzo, C. Herrera
Universidad de Concepción
Camino Einstein Km 2,5., Casilla 4036, Correo 3, Concepción, Chile

W. Alef, H. Rottmann
Max-Planck-Institut für Radioastromonie
Auf dem Hügel 69, 53121 Bonn, Germany

E. Himwich
National Aeronautics and Space Administration/Goddard Space Flight Center
Greenbelt, MD 20771, USA

The requirements for VLBI systems are increasing: higher observation density schedules, real-time access for changing schedules, more automation of observations and remote control of complete sites are being planned. To support these changes new additional software components are required. The addition of (semi-) autonomous, remote accessible control features, which are becoming a reality now will provide needed support by offering reliable, safe, and modular structures from the high-level controlling layers down to the basic equipment interaction elements. An extension to the current NASA Field System (FS) with remotely accessible, autonomous process cells is being developed at the Wettzell Geodetic Observatory. It uses the specially designed middleware generator "idl2rpc.pl", developed at Wettzell, to generate the remote C++- interfaces for communication issues. A new modern graphical user interface in combination with an initial programmatic interface to the FS, both developed as extensions, demonstrate the capability for controlling radio telescopes remotely. The first successful remote control tests, with operators present, during regular experiments with the telescopes at O'Higgins, Concepción and Wettzell have demonstrated that this approach works well in the global communication network.

Science and Technology of Long Baseline Real-Time Interferometry:
The 8th International e-VLBI Workshop, EXPReS09
June 22 - 26 2009
Madrid, Spain

*Speaker.

PoS(EXPReS09)038

1. Introduction

Personnel from the Geodetic Observatory Wettzell operate not only the 20 meter radio telescope at Wettzell. In cooperation with other institutes they also run the 9 meter radio telescope at the German Antarctic Receiving Station (GARS) O'Higgins, Antarctica and the 6 meter radio telescope of the Transportable Integrated Geodetic Observatory (TIGO) Concepción, Chile for geodetic VLBI experiments. Because of the remote locations especially in case of the telescope in the Antarctica a first concept was developed, to control sites remotely on the basis of the current equipment, which is controlled by the NASA Field System (FS) software package. This is a very stable, well known and well supported package. A FS-based extension for new possibilities of remote control and remote attendance was developed. As already existing tools to forward mouse, keyboard and video signals are suboptimal, because they don't allow monitoring of the internet connection itself to facilitate safety actions the new software offers an Ethernet based, safe and stable remote communication. Since most of the new devices controlled by the field system are also connected via ethernet mechanisms the new concept also includes ideas to standardize such individual communication needs. Together it offers the appropriate elements for remote control as a new VLBI observation mode, possibly named "e-control".

2. A principle communication model

The communication in current communication systems on Ethernet is usually request and character oriented (ASCII). This means a service requesting client starts the communication and sends an order request via message communication to a remote, service offering server. At the server side the order is processed and an answer message is returned to the client [3]. This realizes a classic client-server-model.

For the realization of a communication a dedicated computer platform offeres a networking protocol suite. It is organized in form of a protocol stack. Each layer of this stack represents a different level of abstraction from the lower physical media to the higher application part of the communication. In modern platforms mainly protocol stacks on the basis of the Transmission Control Protocol over Internet Protocol (TCP/IP) or the User Datagram Protocol over Internet Protocol (UDP/IP) are implemented. Because of this the newly designed communication sets up on these protocols.

3. The communication with remote procedure calls

In classic communication networks each client-server-interaction is programmed individually during the software development process. This proprietary approach makes it more difficult to set up a remote control and to include or adapt new properties of remote usable devices. Another more modern attempt reduces the efforts of communication programming by defining a standardized way for the transmission of remote procedure calls (RPC). RPCs are comparable to local calls of procedures (functions) in a structured program, but realized as control and data flows over a communication network to allow a standardized interaction between a requesting client and a service offering server [3]. The client just calls a procedure or function without the concrete knowledge

of the processing location, because an additional RPC communication layer realizes the transfer between the client and the remote processing server. The derived answer follows the same way vice versa to the client, so that the procedure call seems as local.

In the given context the Open Network Computing Remote Procedure Call (ONC RPC) is a preferred communication technique because it is available in each Linux operating system as standard and has been well tested since the year 1988. It also includes the platform independent conversion of procedure parameter data using the External Data Representation (XDR) [6]. To reduce the programming effort, the communication layer can be created by using the generator utility "rpcgen". It is also available in each Linux operating system. However it just creates a low-level client-server-communication on the basis of ONC RPC. But for sophisticated remote control implementations higher level techniques, such as connection monitoring on client and server side and safety concepts, are necessary. These add-ons can be used to realize stable states for the hardware even when the communication is broken.

4. The middleware generator "idl2rpc.pl"

Common middleware systems, such as the Common Object Request Broker Architecture (CORBA), extends the described communication with the needed higher level concepts and with advanced services. However these realisations are offered as hugh, bulky, abstract packages. The offered releases are correlated with operating system versions and are often commercial products. Secondary they are not always flexible enough to be used in heterogenous networks with different security guidlines and fiewall implementations because of the additional, proprietary protocol layers.

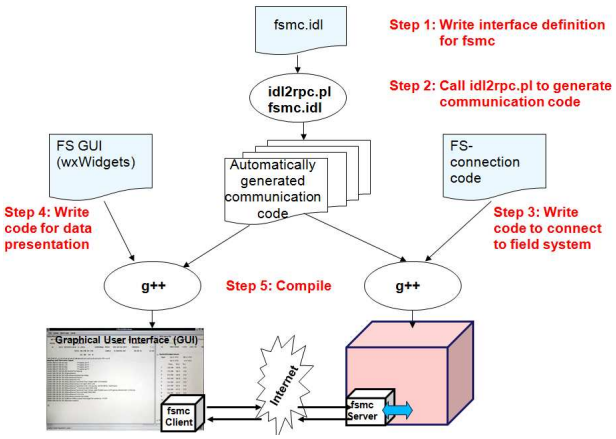

Figure 1: The generation process using the code generator "idl2rpc.pl"

Because of that a new middleware generator "idl2rpc.pl" was designed. This generator itself is just a single Perl script which uses only the ONC RPC realisations of standard Linux distributions. It extends "rpcgen". Several C++ adaptor classes for the C written RPC communication are created. They can directly be used in the application code. Dedicated modules offer threads to use in parallel tasks with semaphores to protect critical sections. A sophisticated communication control mechanisms, such as a watchdog process, which e.g. always restarts the server after an

unexpected crash, extends the stability. An Automatic Safety Device (ASD) controls the existence of a responsible client and forces safety actions when the connection to the client breaks down[2].

"idl2rpc.pl" reads an interface definition file and produces all of the necessary modules. In only a few steps the complete communication is generated with this light generator (see fig. 1). It is open source and no additional, external packages are needed, for example after an upgrade of the operating system. Because of the usage of TCP/IP and UDP/IP without additional, proprietary layers the generated communication can easily tunnel firewall barriers.

This helps to develope distributed systems, consisting of several independent computers (processors), which are connected together to solve a collective task in a cooperative way. During the processing time they don't share memory, clocks or other hardware and just communicate information while transferring messages via a computer network ([3] and [4]). Each hardware device can be represented as server which is accessible via a generated communication. Other servers can contain several clients to access and autonomously control different, subordinated devices. These combination servers themselves again offer services on a generated communication. In summary a hierarchical architecture is realized which establishes hierarchical and autonomous control zones. On the upper end of this hierarchy an user interface allows human interaction.

All of the servers on one platform can be found via registrations in a "portmapper", where all services and their ports are registered. It is also possible to directly contact the servers on the dedicated ports. But the servers can also be distributed on several platforms. The generation method therefor offers the basic and very flexible skeleton to program the different elements of this distributed system, where clients and servers can be individually defined.

5. The layers of e-control as extension of the NASA field System

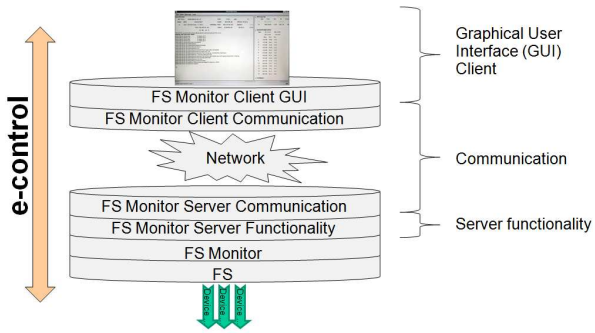

Figure 2: The complete e-control stack

The given remote control design realises a classic client-server-model. In the case of the FS an additional, automatically created communication is defined using the new idl2rpc.pl. The client-server-based complete stack for e-control from user interaction via remote procedure call communication to the FS interaction is shown in fig. 2. One of the main drivers in the given design is a strict separation of control, communication and presentation logic. The complete arithmetic and workflow control logic reside in the server, defined as device control code. It is an autonomous working process which interacts with the remote controlled device (here at this level, the FS). The communication code independently connects the server to the outer world for requesting clients.

The clients are only used to realize an user interface with a presentation of the server processed elements.

In the first implementation the communication consists only of a few methods returning the local information output as string arrays to the remote requesting client. Therefore an additional C-written adapter (FS monitor) allows the connection to the FS via shared memory access. The client can also send a string command to the server. The commands are injected into FS using the supported injection methods. To smooth the communication behaviour the server uses threads to separate between the asynchronous remote procedure calls and the contact with the FS. Semaphore protected variables allow the handling of critical sections when both tasks work in parallel with the same variable values.

Overall the created server acts completely autonomously. It can be used to check system status information independently and can decide what to do to keep stable and safe states. Such controlling utilities can be defined as autonomous process cells. The generated watchdog process keeps it alive and an automatic safety device allows to register if a responsible client is connected. After a breakdown of the communication to the client the server can operate completely autonomously until a critical situation (e.g. increase of wind speed to a level which is critical for the telescope) forces it to run into a safe state. In combination with additional monitoring information around the site (meteorology, power supply, air conditioning status, etc.) that compact server extends the FS for a reliable remote control.

6. The new graphical user interface for remote control of the NASA field System

On the other end of the remote control the operator interacts with the system. It is possible to implement command line clients as well as highly sophisticated web applications or graphical user interfaces. This permits management of devices remotely via browser, command line and/or graphical user interface (GUI). For the described FS extension wxWidgets is the prefered way to offer a GUI. wxWidgets is a C++ based open source framework for platform independent developments of graphical user interfaces [5]. Although the current RPC generator only supports Linux systems (32 and 64 Bit), the graphical user interface is modular enough to support different platforms like Windows, Linux, OSX and others. In terms of the proposed FS extension a new graphical user interface was created using the wxWidgets framework for its realisation. To keep usage similar, the display elements are organized to be like those of the current local interface to the FS.

7. Safety and security in the current implementation

To protect humans and the system itself safety and security concepts are currently in development in addition to the inline safety functionality of the generated communication. Safety hereby means the protection against local error states or critical situations, possible for automatically moving hardware. An additional, modular and multi-layered system monitoring hardware is in production which should check all of the important system parameters. Security however means the protection of the system from not allowed activities done by foreign attackers or users without the sufficient right policies. To bring in such a level of security the Secure Shell (SSH) tunneling methods with its several authentication possibilities can be used to build up an access protection. For

the internal correct access for operator actions it is planned to realize an authentication (registration of a user with username and password) and authorization (personification of a user for a specific remote procedure with dedicated rights) [1].

8. Remote control tests

To prove the functionality of the remote control and the general character of the implementations as well, several tests were initiated to run the radio telescopes operated by Wettzell with the described software. Several 24 hour and 1 hour intensive experiments were successfully run by remote control also at the very remote site O'Higgins. These tests will be extended and will lead to routine remote operation of VLBI experiments at Wettzell.

9. Summary and outlook

Overall the described method allows the development of distributed systems consisting of several independent servers which act completely autonomously. It extends given structures to have a remote control possibility and splits complex systems up into several manageable units interacting together with a general, standardized but also flexible communication method.

The resulting software is an option for upcoming Fundamental Stations with several different colocated measuring systems like radio telescopes and laser ranging systems to realize remotely controllable, autonomous subsystems especially along the goals of the Global Geodetic Observing System (GGOS). New observing strategies proposed in VLBI2010 can be realised also with very remote stations. But nevertheless there are always some situations which cannot be controlled and handled by such an automated system (like access locks after power failures), so that responsible, well educated engineers at the sites should always be the final instance of automation.

References

[1] A. Neidhardt, *Verbesserung des Datenmanagements in inhomogenen Rechnernetzen geodätischer Messeinrichtungen auf der Basis von Middleware und Dateisystemen am Beispiel der Fundamentalstation Wettzell*, Mitteilungen des Bundesamtes für Kartographie und Geodäsie **37** (2006).

[2] A. Neidhardt, *Manual for the remote procedure call generator "idl2rpc.pl"*, Geodetic Observatory Wettzell (2009).

[3] M. Singhal, N. G. Shivaratri, *Advanced Concepts in Operating Systems*, McGraw-Hill, Inc. (1994).

[4] A. Puder, K. Römer, *Middleware für verteilte Systeme*, dpunkt-Verlag GmbH (2001).

[5] J. Smart, K. Hock, S. Csomor, *Cross-Platform GUI Programming with wxWidgets*, Prentice Hall International (2005).

[6] R. W. Stevens, *Programmieren von UNIX-Netzen. Grundlagen, Programmierung, Anwendung* Prentice-Hall International, Inc. (1992).