

## EGU2010 SM1.3 Seismic Centers Data Acquisition session

# The EGU2010 SM1.3 Seismic Centers Data Acquisition session: an introduction to Antelope, EarthWorm and SeisComP, and their use around the World

Damiano Pesaresi<sup>1,2,\*</sup>

<sup>1</sup> Istituto Nazionale di Oceanografia e di Geofisica Sperimentale – OGS, Udine, Italy

<sup>2</sup> Istituto Nazionale di Geofisica e Vulcanologia – INGV, Rome, Italy

## Article history

Received May 5, 2010; accepted February 24, 2011.

## Subject classification:

Surveys, measurements, and monitoring, Data processing, Seismological data.

## ABSTRACT

Session «SM1.3 – Seismic Centers Data Acquisition» at the General Assembly 2010 of the European Geosciences Union (EGU), taking place in Vienna (Austria) between 2–7 May 2010, was organized to present both differences and similarities in operations by different types of seismic data centers in order to share experiences and stimulate constructive discussions. Only a few, widely used, "all-in-one" data acquisition and processing packages are available for seismic data centers, two public domain tools (SeisComP and EarthWorm) and one commercial tool (Antelope). The choice for any particular tool may depend on many different criteria, from operational aspects to scientific results, or on the availability of specific requirements in relation to a specific mission. The development of EarthWorm originally started in 1993 in the USA to replace aging and vendor tied regional processing systems. Antelope, on the other hand, started around 1996 with the aim to have real-time data flow from the field sensors to the scientist. SeisComP also started in the nineties as real-time data acquisition and processing system and evolved initially towards an early warning system for seismic observatories. Protocols have been established to exchange real-time waveform data between the different packages.

## 1. Introduction

Global, regional and local seismic data centers today differ significantly in scope, size and type of operations. While the common goal is to collect, archive and process data for the purpose of earthquake monitoring and seismological research, both mission and resources usually determine the scale and kind of operations.

The increased number of seismic monitoring stations as well as the increased availability of real-time seismic waveform data today, put high demands on software and seismic data centers in terms of acquisition, processing and quality control. The large amount of data and the demand for real-time analysis can only be handled with automated, but flexible, software systems being reliable, robust and sustainable. At

present the seismological community basically has the choice between three systems (SeisComP, EarthWorm and Antelope) to fulfill the needs of any seismic monitoring system.

The choice for a particular system usually depends on the specific goal(s) of the monitoring system, local resources, knowledge and sustainability. Items that may be considered in the evaluation of different software tools for seismic data acquisition are (listed randomly):

- availability and extent of documentation
- support and maintenance
- costs (commercial, open source, license)
- user's group community and support
- coordination in developments and contributed software
- installation and implementation
- datalogger support (data acquisition)
- data exchange (supported formats and protocols)
- data verification and completeness (quality control monitor)
- system configuration (adding stations and metadata)
- error checking and error reporting
- implementation of external software modules (add-on's, plugins)
- implementation of other/new algorithms
- implementation of other velocity model(s)
- GUI
- automatic processing (picking, association, location, magnitude)
- manual processing
- reproducing event parameters from previous analysis system
- data management, archiving (db) and access.

## 2. Antelope

At the end of the IRIS Joint Seismic Program (JSP), a decision was made to develop a new "Broadband Array" facility

within the PASSCAL program using experience, equipment and technologies that were acquired during the life of the JSP. The central objectives of the new facility were to provide in real-time high resolution digital data from the field recorders directly to the PI's or other interested parties and to give open access to the data through IRIS DMC (<http://www.iris.washington.edu/>) as quickly as possible (within several days). It was hoped that the new Broadband Array facility could act as a prototype for future PASSCAL field deployments and could start to give the research community experience with the new communication and data processing technologies that would become available within the new facility.

This software evolved from the Datascope Seismic Application Package (DSAP), a public-domain seismic data management and processing suite that was developed at the University of Colorado, Boulder, with funding from the IRIS JSP. The main shortcoming of DSAP, however, was also the main objective of the Broadband Array: real-time data flow from the field sensors to the scientist. Through a series of meetings, system specifications, system design, prototype implementations and system evaluations, a new software facility, known as the Object Ring Buffer (ORB), was ultimately developed at the University of Colorado, Boulder, to support the initial testing and operational phases of the PASSCAL Broadband Array. The first ORB software was tested with live data starting in the summer of 1996 from the initial test deployment of the Broadband Array equipment in Southern California. The first operational use of the ORB software started in the summer of 1997 after the deployment of the Broadband Array in Northwestern Colorado for the Lodore Array Project.

In the fall of 1996, Boulder Real Time Technologies, Inc. (BRTT) was formed as a Colorado corporation. BRTT's commercial mission was to provide technical expertise and computer software to support the operations of seismological networks worldwide. Shortly after its formation, BRTT entered into a contract with Kinematics, Inc.

In the winter of 1997 BRTT decided to address all of the problems and weaknesses that were uncovered in the evaluation of the public-domain implementation. It was determined that the most efficient way to do this would be a complete rewrite of all of the ORB software. For the Datascope software, the table locking was made as a modification to the public-domain implementation. In order to do this right, BRTT developed a formal ORB test suite that was used to flog the system as it was developed. BRTT also started taking live data feeds from several networks that were using the public-domain Datascope/ORB software. This revealed bugs and other system performance problems that were sorted out as the new ORB was developed. The basic concepts and the software interfaces in the public-domain Datascope/ORB were, for the most part, preserved in the commercial version.

By late summer 1997 a working prototype of the new commercial ARTS system was being tested. This new system included completely rewritten ORB software, revised Datascope software and a suite of new ORB and Datascope client-application programs to support automated real time seismic network processing. At about the same time BRTT met with IRIS and proposed to use the commercial Antelope software to run the Broadband Array deployment in Northwestern Colorado. In fall of 1997 BRTT and IRIS entered

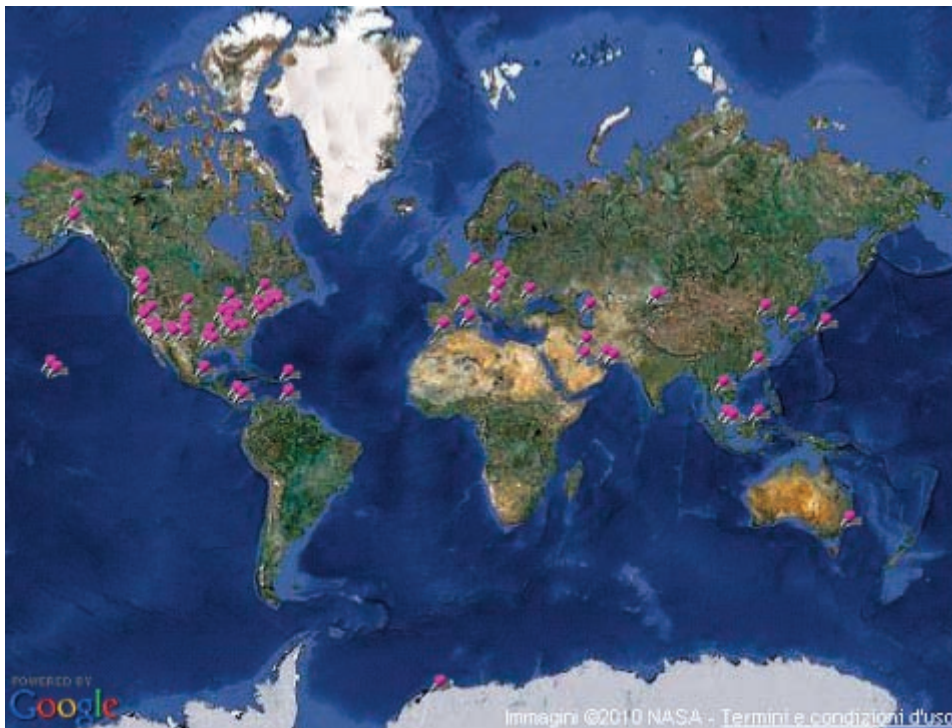


Figure 1. Antelope distribution (status: 2011).

into a license agreement which provided IRIS limited use of the commercial Antelope software. The IRIS Broadband Array deployment in Northwestern Colorado started using the commercial Antelope software in the fall of 1997. Up until this time there had been two development tracks for the Datascope/ORB software; the public-domain track and the commercial track. After the fall of 1997, development of the public-domain version of the Datascope/ORB software ceased (with the exception of several PASSCAL related programs for dealing with SEED data). The final version of the public-domain Datascope/ORB software was delivered to IRIS in June 1998 and is available, including all source code, on IRIS' anonymous ftp site at ftp.iris.edu.

The commercial ARTS software has been under continuous development since fall 1997. In summer 1998 the IRIS PASSCAL program contracted BRTT to continue support of Antelope software for all PASSCAL experiments, including the experiments using the Broadband Array facilities [BRTT 2011].

### 3. EarthWorm

The EARTHWORM project was started in 1993, mainly in response to a number of common needs identified by regional seismic networks in the USA: (1) the automatic processing systems used by most regional networks were aging, and maintenance costs were growing; (2) advances in seismic research required data from new, sophisticated dataloggers and sensors in order to provide valuable research data; (3) growing societal demands for new, immediate available and highly visible real-time products and (4) decreased funding by which most individual networks could no longer support or maintain local system developments.

Most of the processing systems then in use had a number of drawbacks which made it difficult to enhance them to meet the new requirements. They were built on vendor-specific products, which tied them to high-price vendors, and deprived them of the benefits of the rapidly growing mass market. They tended to be tightly integrated, in the sense that functionally unrelated parts of the system shared a common resource, such as processor cycles, internal data paths, or peripheral devices. This resulted in systems that were difficult to modify and maintain, as changes to one function could cause malfunctions in an unrelated area.

Since the initial objective of this system was to provide rapid notification of seismic events, the system that evolved had no "memory" of past events. The emphasis was on speed and reliability, and an event was completed when the system produced a notification. As the project progressed, however, additional needs were requested such as interactive review of acquired events, association of late-arriving data, incorporation of strong motion data, and production of catalogs and archive volumes.

These new requirements implied the need for data

storage and user interactions. Given the overall design goals (below), a commercial DBMS was chosen as the storage mechanism, and dynamic web pages were chosen to implement the user interaction. This represented a significant shift from the original orientation of the project: a focus on carefully controlled code and minimal use of third-party software packages. The new requirements however, geared the software into complexity and reduced real-time performance. In order to not compromise the original goal of speed and reliability, it was decided to separate the project into two areas — the Automatic Earthworm and the Interactive Earthworm. The Automatic Earthworm is a continuation of the original real-time effort. Development and support continues within its original objective of providing rapid, reliable notification. It can be compiled and run as before, requiring only compilers and basic operating system services. Configuration is conceptually fairly simple, and maintenance requirements are minimal.

The Interactive portion, however, requires a DBMS (currently Oracle) with its associated interface codes, a web server, and usually, dedicated computers. Configuration and maintenance requirements are considerably higher than for the Automatic system. The benefit of this complexity is that it permits a new level of services and interaction modes. A number of Earthworm versions providing such features have been released, and the bulk of current development work is in this area [USGS 2001].

#### 3.1. Earthworm design goals

**Modularity** — Each function performed by the system should be encapsulated into a module that can work independently of other modules, in terms of hardware as well as software. The implication is that a set of critical system functions can be guaranteed to be independent of other functions in the system. Thus, new, experimental functions can be added without disrupting pre-existing critical operations. Also, quality assurance can be performed on a module-by-module level, rather than for the entire system. This has proved to be a difficult principle to adhere to in practice. First, one has to resist the temptation to use intermediate results already computed by other modules and second, to resist the lure of using various common utilities which would streamline operations.

**System independence** — Such modules should operate on various brands of computer hardware and operating systems, and various types of such computers can be linked together to operate as one system. This, with the idea of modularity above, implies that the system can gradually migrate from one type of computer to another without disruption. In practice this means the use of only standardized portions of various computer systems, and isolating any unavoidable system specific functions in standardized wrapper routines.

Scalability — The system is developed to provide a smooth cost-performance curve to accommodate large as well as small networks. This was perhaps more significant prior to the availability of cheap, powerful mass-produced computers. However, it is still relevant in regard to license fees for included commercial software.

Connectivity — The assumption is that such systems are no longer isolated, but have to interact quickly and reliably with other automatic real-time systems, interactive analysis systems, and various notification schemes. The objective is to provide connectivity at various levels of automatic and interactive processing, ranging from stand-alone configuration to a distributed, networked configuration.

Robustness — Traditionally, the role of seismic processing systems was mainly one of research support. As a result, system reliability was naturally less important than cost, processing time, and specific research features. However, the new responsibilities to the press and emergency response agencies required very high levels of system reliability, especially during seismic crises, when input data and power may be interrupted, and system loads may increase dramatically. Thus issues like error detection and recovery, time-to-repair, graceful degradation, and load control were becoming vital, and had to be designed into the system. The costs of robustness are often not appreciated, in terms of time to design, implement, and test, as well as costs for suitable hardware and additional equipment.

### 3.2. Automatic Earthworm architecture

An EARTHWORM system consists of a set of modules "immersed" in a message passing "medium". Each module performs some coherent task, such as data acquisition, phase picking, etc. Modules communicate by broadcasting and receiving various messages such as packets of trace data, phase picks, etc. The message passing scheme is analogous to radio communications: It consists of a message-carrying "medium" and a set of standard routines which can be thought of as multi-frequency two-way radios operating in this medium. Modules can use these routines to broadcast messages into this medium on a specified "frequency", and to "tune in" to messages on specified "frequencies". Significant properties of this scheme are that: (1) a module cannot be prevented from broadcasting a message whenever it chooses; (2) any number of modules can listen in to some sending module(s) without affecting that module; (3) a module receives only the messages of interest to it and it is notified if it missed a message.

This "medium" (the EARTHWORM transport layer) operates within one computer as well as between different computers. Standard UDP/IP network broadcasts are used to carry a batch of messages between computers and shared memory regions are used to simulate such broadcasts within a computer. Each participating computer can be equipped

with multiple network ports and multiple shared memory regions. Special adapter modules are used to link the message flows between a shared memory region and a network cable. Thus an EARTHWORM system can be configured to have a geometry of message passing paths tailored for a specific application, operating on any number of computers. Note that this scheme operates within one EARTHWORM system, spanning a maximum distance of several hundred meters. Message exchange between remote EARTHWORMs is handled by a different mechanism. The selective "frequencies" of this "medium" are implemented by attaching "shipping tags" to all messages. They specify the module which created the message, the type of message it is, and the installation where the EARTHWORM system is running. Thus, for example, a message may be labeled as being a p-pick produced by the Rex Allen picker at Menlo Park, or a location produced by Hypo-inverse at Seattle. These tags are then used by the receiving routines to provide only the desired messages to a listening module.

The modules are independent main programs. Each module has free use of the file system and other system facilities. However, the EARTHWORM system offers several suites of support routines that can be used by the program to make it a compliant EARTHWORM module. One such suite is the collection of message passing routines (transport routines). These are used to acquire input data by requesting messages of certain type(s), and to broadcast output messages. Another set of routines is used to effect operating system independence by providing a common set of "wrapper" routines for system-specific functions. For example, the system routines for starting a thread are different on Solaris and NT; the EARTHWORM system provides routines for both systems with the same name and arguments. Thus, if a program uses that routine, it can run on either operating system: the proper system-specific version of that routine will be automatically inserted when the program is built. This suite currently contains such "wrappers" for all common functions. Routines are added as required. Moving EARTHWORM to a new operating system involves creating a new set of such "wrapper" routines. EARTHWORM currently offers "wrappers" for OS/2, Solaris, and Windows NT.

Modules generally read a configuration file at startup time. This file contains basic EARTHWORM parameters which define the module, specify which message "medium" (message ring) it wishes to listen to, and which message ring it wishes to broadcast its output messages to. In addition, this file can contain arbitrary operating parameters specific to its operation.

Error detection is accomplished via several mechanisms: Any module may broadcast an error message. Such messages are forwarded to a special module (StartStop, below), which takes appropriate actions based on the nature of the

complaint. Each module can also provide a periodic heartbeat; if this heartbeat should cease an error would then be declared and processed. A module may also request to be restarted if its heartbeat should cease. In addition, the EARTHWORM as a whole can produce a heartbeat to an external monitoring device. Such a monitoring device, capable of generating pager messages, is available as part of the EARTHWORM system.

### 3.3. Interactive Earthworm architecture

The DBMS "instance". — In 2001 this was an Oracle, Standard Edition, version 8, replaced then by SQL. It usually runs on its own computer, and offers a network-based service to which applications can attach, request data transfers, and then detach. Much like multi-user operating systems, such DBMS "instances" can support many concurrent, independent users. Thus, Interactive Earthworm does not require its own physical "instance"; it can operate within a department-wide or remote instance.

The schema. — This is the set of data base tables which reside in the DBMS instance. This consists of the data values to be stored, and the scheme of linkages between tables which form the associations between data. The schema currently consists of several sub-schemas for storing event, trace, response, and alarm data.

The DBMS API. — These allow application programs to access the data in the schema.

Real-time "stuffer" modules. — This is a suite of coupler modules which run as part of the automatic real-time system. They attach to real-time message rings and listen for specified messages carrying real-time results, such as location or magnitude estimates. On the output end, coupler modules attach to a DBMS, and insert such data via the API routines.

A web-page server and an associated suite of application programs. — This subsystem is used to support most user interactions. A user can access the system via any machine equipped with a suitable web browser, and will be served a variety of web pages. As the user makes selections via buttons on these pages, the web server invokes the associated application programs which connect to the data base, and compute new web pages based on the data in the DBMS. User selections on such web pages, in turn, can be interpreted by these programs, and be used to alter the data in the DBMS.

### 3.4. EarthWorm DBMS API

The DBMS API mentioned above is of special significance. This consists of a set of C-callable functions which can be called by any program in the project. They provide a user-oriented view of the data base via calls such as CreateEvent(), GetEvent(), CreateMagnitude(), GetMagnitude() etc, and via a suite of associated data structures. These routines, in turn, link to the DBMS server and execute associated SQL procedures stored in the server. It is these

---

## EartWorm installations around the World (2010)

---

1. Alaska Volcano Observatory (AVO)
  2. Alaska Tsunami Warning Center (ATWC)
  3. British Geological Survey (BGS)
  4. California Integrated Seismic Network (CI/NC/BK/NP)
  5. Cascade Volcano Observatory (CVO)
  6. CSIRO Exploration and Mining, Australia
  7. Dominican Republic
  8. Ecuador (IGEPN)
  9. El Salvador (SNET)
  10. Hawaii Volcano Observatory (HVO)
  11. Hellenic Seismological Broadband Network (HL), Athens, Greece
  12. Idaho National Laboratory (INL)
  13. Infrasound Laboratory, University of Hawaii (ISLA)
  14. Instituto Nazionale di Geofisica e Vulcanologia (INGV), Italy
  15. Institut Geologic de Catalunya, Catalonia, Spain
  16. Kandilli Observatory, Turkey
  17. Lamont-Doherty Cooperative Seismographic Network (LD)
  18. Monserrat Volcano Observatory
  19. Montana Regional Seismographic Network (MT)
  20. Mount Erebus Volcano Observatory
  21. New England Seismic Network (NESN)
  22. New Mexico Tech Seismic Network
  23. National Earthquake Information Center (NEIC)
  24. New Mexico Tech Earthworm System
  25. New Madrid Seismic Network, CERI
  26. Nicaragua Seismic Network Survey (INETER)
  27. National Institute for Occupational Safety and Health (NIOSH) Mining Division
  28. Norwegian National Seismic Network (NNSN)
  29. Volcan Baru (OSOP)
  30. Observatorio Vulcanologico y Sismologico de Costa Rica (OVISCORI)
  31. Pacific Northwest Seismic Network (PTWC)
  32. Pacific Tsunami Warning Center
  33. Patras Seismological Laboratory (Greece)
  34. Puerto Rico Seismic Network
  35. Servicio Seismologico Nacional (Mexico)
  36. University of Colima (UC)
  37. University of Utah Seismographic Stations (UU)
  38. University of Panama
  39. Uzbekistan Digital Seismic Network
  40. USGS Geomagnetic Program
  41. Venezuela Seismic Network (FUNVISIS)
  42. Autonomous University of Santo Domingo (UASD), Dominican Republic (DR)
- 

**Table 1.** EartWorm installations around the World (2010).

SQL procedures which alter the contents of the DBMS.

They shield the applications author from the complexity of the schema and DBMS protocols. The API calls are written to accept data structures meaningful to typical applications, and perform the translation to schema tables via procedures stored in the DBMS server.

They protect the DBMS data from corruption. If applications were to directly access DBMS tables, there is the possibility that an ill-behaved application could corrupt the data base. The critical access routines are stored on the DBMS server machine, and are maintained by the same mechanism as the schema.

They provide for portability. Since the applications are unaware of the schema and the DBMS primitives, changes to the schema — and indeed to the DBMS vendor — can be absorbed by changes to the API.

Provide control of connections to the DBMS.

#### 4. SeisComP

SeisComP is one of the most widely distributed software package for seismological data acquisition and real-time data exchange over Internet [SeisComP3.org 2011a]. Its data transmission protocol SeedLink became a *de facto* world standard. The first version of SeisComP was developed for the GEOFON network and further extended within the MERIDIAN project under the lead of GEOFON/GFZ Potsdam and ORFEUS. Originally SeisComP was designed as a high standard fully automatic data acquisition and (near) real-time data processing tool including quality control, event detection and location as well as dissemination of event alerts. In the context of the GITEWS project (German Indian Ocean Tsunami Early Warning System) additional functionality were implemented to fulfill the requirements of 24/7 early warning control centers. Major changes in the architecture

of SeisComP were necessary and many new features result in the upgrade of SeisComP to version 3.0. Important SeisComP releases are shown in Table 2. A first prototype of SeisComP3 developed by the GITEWS/GEOFON development group was released in May 2007. SeisComP3 provides the following features:

- data acquisition
- data quality control
- data recording
- real-time data exchange
- network status monitoring
- real-time data processing
- issuing event alerts
- waveform archiving
- waveform data distribution
- automatic event detection and location
- interactive event detection and location
- event parameter archiving
- easy access to relevant information about stations, waveforms and recent earthquakes.

The new requirements for early warning purposes made it necessary to adapt the design and architecture of the previous SeisComP. The guidelines for the design of SeisComP3 are:

- implementation of critical functions as standalone modules to guarantee the independence from other functions (e.g. picker, magnitude calculation, interactive analysis)
- easy implementation of custom modules
- independence of hard- and software
- ability of data exchange between different automatic real-time systems
- distribution of modules on several systems
- robust system for rapid and reliable earthquake solutions (especially during seismic crises).

Version	Date	Function/Module
1.0	February 2001	SeedLink 2.0 (plug-in interface) Plug-ins for EarthData PS2400 and Lennartz M24
1.1	August 2001	SeedLink 2.1 (streams.xml, improved buffer structure) make conf/make key scripts LISS plug-in, SeedLink-Antelope connectivity
1.1.5	January 2002	SeedLink 2.5 (multi-station mode)
1.16	March 2002	GIF live seismograms
2.0	October 2003	SeedLink 3.0 (INFO request, time-window extraction) libslink, chain plug-in, Comserv independence
2.1	June 2004	Python add-on package (SeisPy) including AutoLoc2 chain plug-in extension interface, triggered streams
2.5	March 2006	Integration of add-on packages, modular config. script
3.0 alpha	May 2007	New architecture, new magnitude types, GUI
3.0 release	May 2008	Stability and performance improvements, improved GUI functionality

**Table 2.** SeisComP versions.

4.1. SeisComP architecture

A SeisComP3 automatic system consists of a set of independent applications each performing a discrete task. The communication between the applications is realized by a TCP/IP based messaging system. This messaging system is based on the open source toolkit "Spread" that provides a high performance messaging service across local and wide area networks. At the top of "Spread" a mediator, called scmaster handling additional requirements of SeisComP3 that are not natively provided by "Spread". The messaging system is used for the exchange of meta data (e.g. picks) and administration of the program modules. The data model of SeisComP3 is based on the QuakeML schema version 0.5. QuakeML is also used as database object schema. By default SeisComP3 uses a MySQL database, but PostgreSQL is supported too. Figure 2 shows a simplified SeisComP3 system.

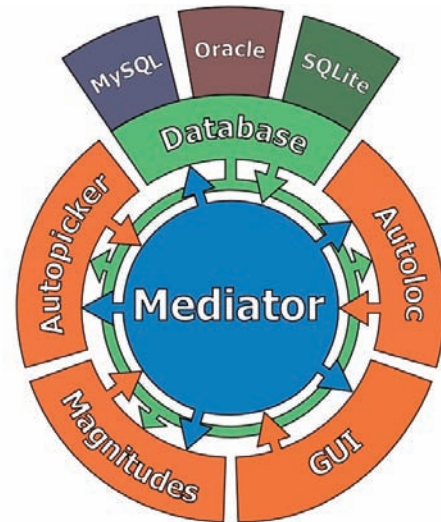


Figure 2. SeisComP3 System.

The waveform data acquisition is based on the well established SeedLink protocol and the new ArcLink protocol both developed at the GFZ Potsdam. The applications in SeisComP3 can be divided in four different groups: data acquisition, processing, graphical user interfaces and utilities.

One requirement of SeisComP3 was to minimize database (DB) access. Therefore most of the applications have an internal cache storing objects received from the messaging. One rule which is consequently implemented is that SeisComP3 applications generally have only read access to the DB. The only application with writing permission to the DB is scmaster. Scmaster receives messages determines the receiver group and forwards the messages similar to a mailman. If instructed, scmaster also writes the included object to the DB. All applications communicate only via the messaging system. For communication three types of messages are used. The "data message" contains plain objects like QuakeML objects (Figure 3).

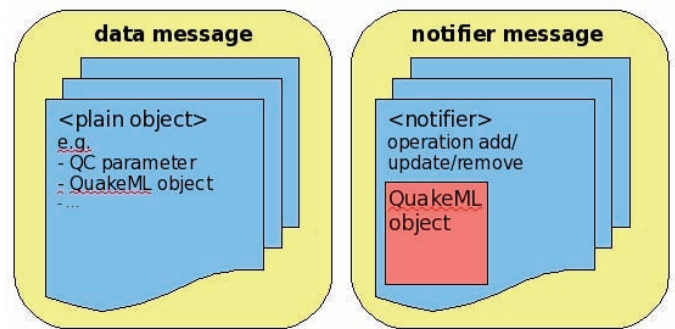


Figure 3. SeisComP3 schematic data and notifier message.

Normally this message type is used for non persistent information. The "notifier message" is the primary message used for QuakeML object exchange between the different applications. It contains a data object (e.g. a QuakeML object) and an instruction what to do with this object. Available instructions are add, update and remove. Scmaster follows these instructions and applies it to the DB. For example in case of an add instruction it adds the included objects to the DB. All message types are additionally classified to messages groups that define the receivers of the message. Such message groups are for example PICK or LOCATION. The third message group is responsible for SeisComP3 administration called "service messages" [SeisComP3.org 2011b].

5. Conclusions

The session «SM1.3 – Seismic Centers Data Acquisition» at the General Assembly 2010 of the European Geosciences Union (EGU), which took place in Vienna (Austria) between 2–7 May 2010, had 23 presentations, 7 of which were selected as oral and 16 as posters. They represented the usage of

Antelope, EarthWorm and SeisComP seismic acquisition software tools all around the world, including USArray and IRIS DMC in the USA, South America, ORFEUS, OGS and others in Europe, Asia.

**Acknowledgments.** The author wishes to thank the European Geosciences Union for supporting the organization of the session «SM1.3 – Seismic Centers Data Acquisition» at the General Assembly 2010, and the Annals of Geophysics for believing and supporting its publication.

References

BRTT (2011). Evolution of the Commercial ANTELOPE Software; <http://www.brtt.com/docs/evolution.pdf>.  
 SeisComP3.org (2011a). Introduction and scope; <http://www.seiscomp3.org/wiki/doc/introduction-and-scope>.  
 SeisComP3.org (2011b). Software architecture; <http://www.seiscomp3.org/wiki/doc/software-architecture/>.  
 USGS (2001). EarthWorm Overview; <http://folkworm.ceri.memphis.edu/ew-doc/>.

\*Corresponding author: Damiano Pesaresi, Istituto Nazionale di Oceanografia e di Geofisica Sperimentale – OGS, Dip. Centro di Ricerche Sismologiche (CRS), Udine, Italy; email: [dpesaresi@inogs.it](mailto:dpesaresi@inogs.it).