Open Software Tools for Reproducible
Computational Geophysics Workshop –
Houston
16-17 June 2011

SeaSeis – A simple seismic processing system

Bjorn Olofsson

Outline

- SeaSeis batch processing system
  - Background
  - System design
  - Strengths & weaknesses
  - Future plan

- Demo
  - SeaSeis processing flows, logs and job submission
  - SeaView – 2D seismic viewer
  - How to write a SeaSeis module

SeaSeis – Background

SeaSeis is a sequential trace flow system for processing seismic data. It has been developed since 2005 by myself, with lots of input from others, and generally from the Internet.

Why SeaSeis?

Subjective

- Available systems at the time lacked user-friendliness and graphical capabilities to properly visualise seismic data
- It's fun!

What does it consist of?
- Batch system managing the trace flow, including flow branching, and internal memory allocation
- Processing flow parser (ASCII text file)
- Command line submission tool
- Around 75 processing modules, including a short online documentation
- Interactive 2D seismic viewer
- Functionality for master flow generation, variable substitution, job submission and log generation

# SeaSeis – Main Focus & Strengths

- Make it as simple as possible at every level, in particular for the user
- Make it easy to compile (no/few dependencies), and easy to write new modules
- Allow automation of repetitive tasks → master flows
- Reproducibility → Logging, error reporting, versioning
- Supply high fidelity 2D seismic viewing capability

SeaSeis – What SeaSeis is NOT

- SeaSeis is NOT equipped with all standard seismic processing tools:
  SeaSeis has currently only a limited number of processing modules. The module base is most complete for onboard QC processing, in particular for seabed seismic data processing. SeaSeis also provides a stable base system in which new modules can be easily added into.
- SeaSeis was NOT written by an IT expert, or Geophysics guru:
  SeaSeis was written by a user frustrated by inefficient, complicated, defect, or overly expensive processing systems. Thus, it lacks shiny examples of super-modern information technology, or extremely complex geophysical algorithms and methods.
- SeaSeis is NOT highly efficient on the computer hardware:
  SeaSeis is intended to be highly efficient mainly for the user, not for the CPU. The lack of a GUI frontend currently hampers its usability somewhat, but this should not scare off users who are good at the keyboard and on the command line.
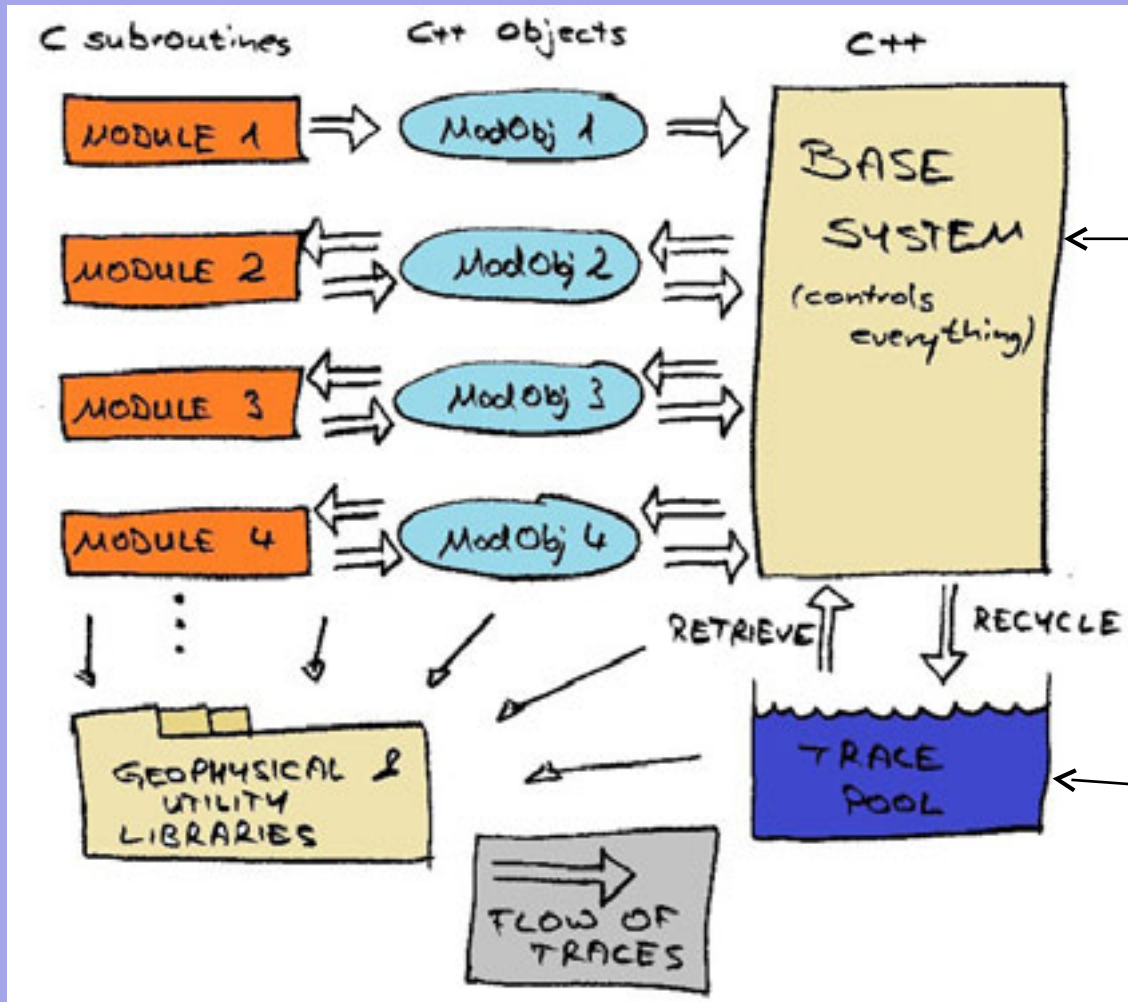
SeaSeis – What SeaSeis is NOT

Processing systems using standard I/O
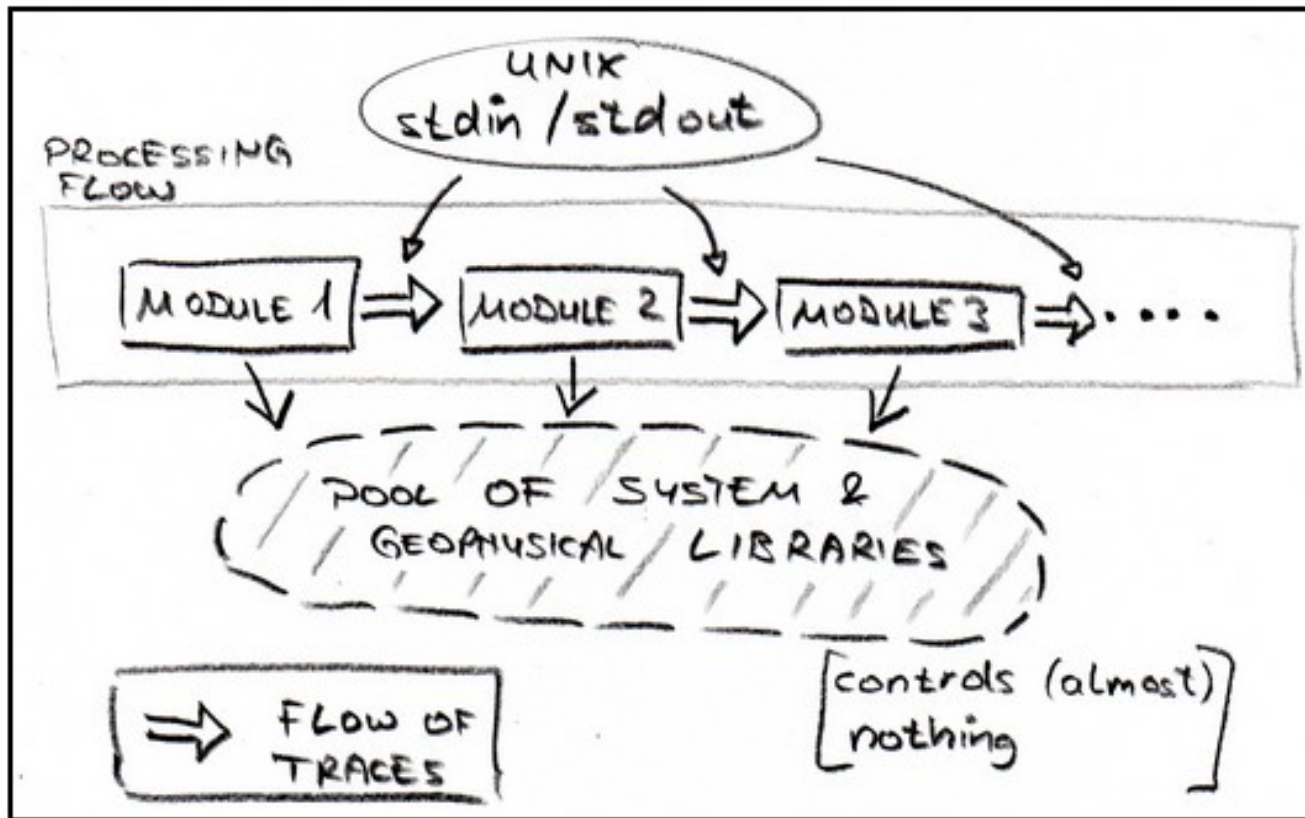
SeaSeis – System Design

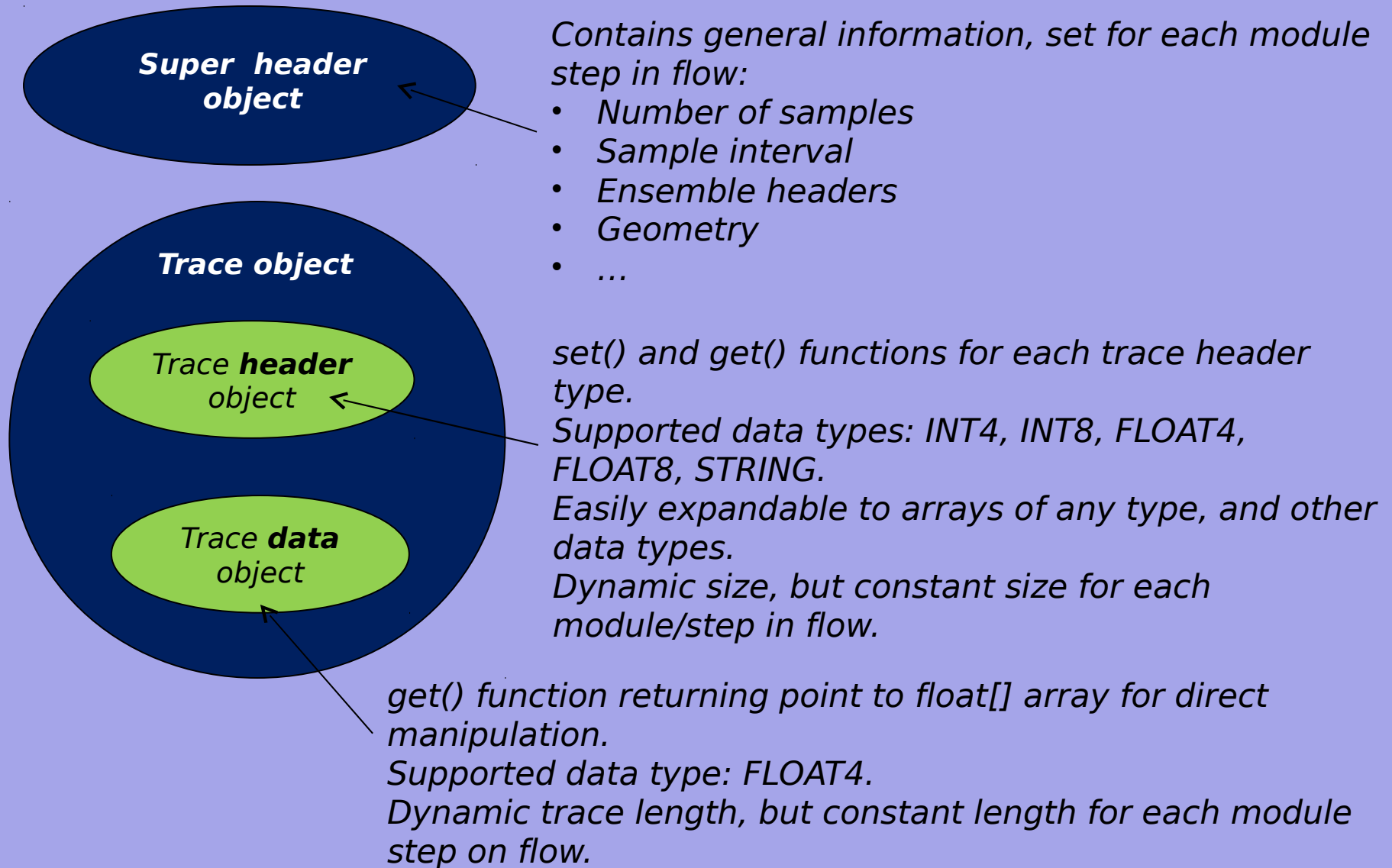- 95% of the SeaSeis batch code written in C++, the remaining part in C

Each module is pre-compiled into a dynamically linked shared object (.so) file.

Trace flow is basically a loop that passes every input trace through all modules

Each trace object is created only once and is passed through the processing flow by reference.

SeaSeis – System Design

Super header object

Contains general information, set for each module step in flow:
- *Number of samples*
- *Sample interval*
- *Ensemble headers*
- *Geometry*
- *...*

Trace object

*Trace **header** object*

set() and get() functions for each trace header type.
Supported data types: INT4, INT8, FLOAT4, FLOAT8, STRING.
Easily expandable to arrays of any type, and other data types.
Dynamic size, but constant size for each module/step in flow.

*Trace **data** object*

get() function returning point to float[] array for direct manipulation.
Supported data type: FLOAT4.
Dynamic trace length, but constant length for each module step on flow.

SeaSeis – Run time

On execution, SeaSeis dynamically loads the shared object files for all modules specified in the processing flow. No further compilation is required.

Each module needs to implement three subroutines:

| Param Method | *Defines all user parameters, including online documentation* |

| Init Phase | *Called once for initialisation, no traces passed.* |

| Exec Phase | *Called once for each input trace or trace ensemble.*
Called one more time for cleanup. |

SeaView – System Design

- SeaView is written in Java, and consists of:
  - Generic 2D seismic display graphics engine (CSeisLib)
  - Access to SeaSeis code through Java Native Interface (JNI), mainly data readers
  - SeaView application: 2D seismic viewer
  - PlotImage application: Presentable image generation

Seismic 2D display written using generic Java 2D graphics library.
Why not OpenGL? → More difficult to code high fidelity wiggle display..??

SeaView – Functionality
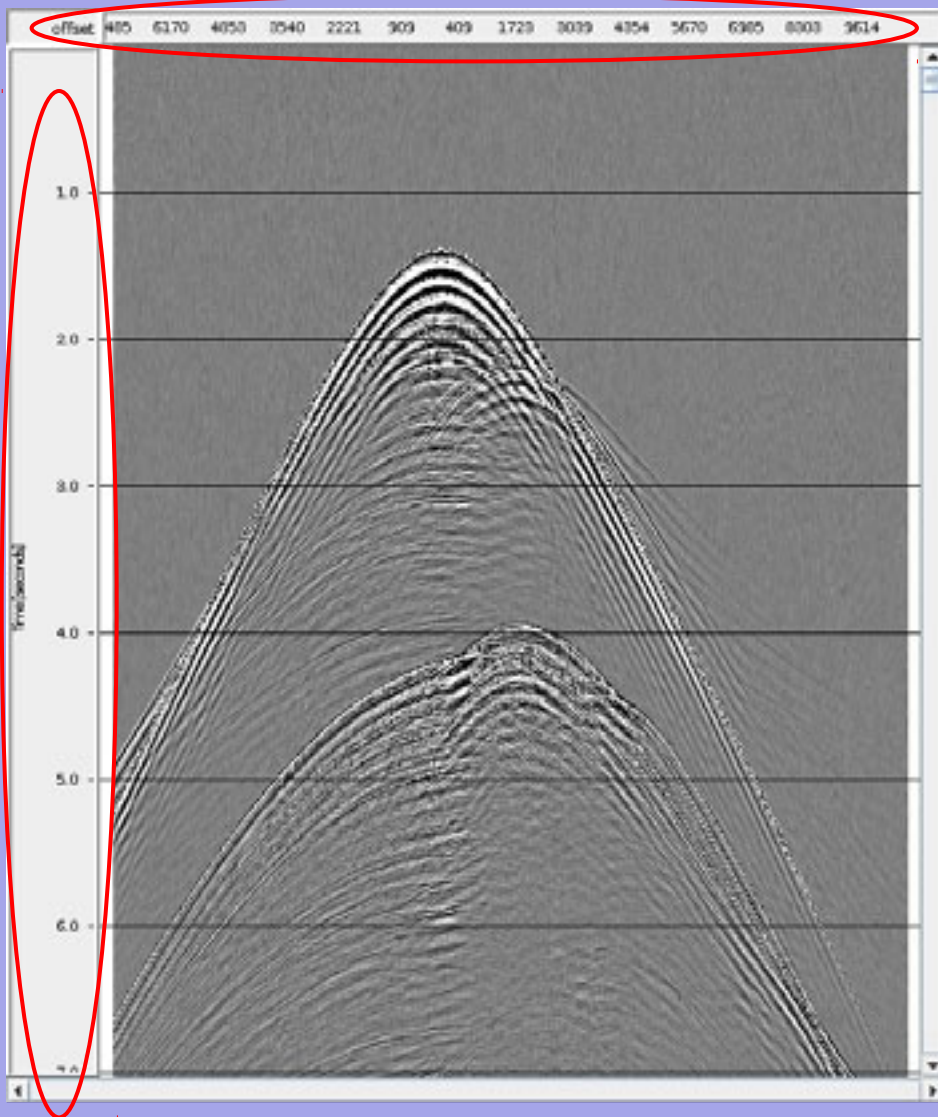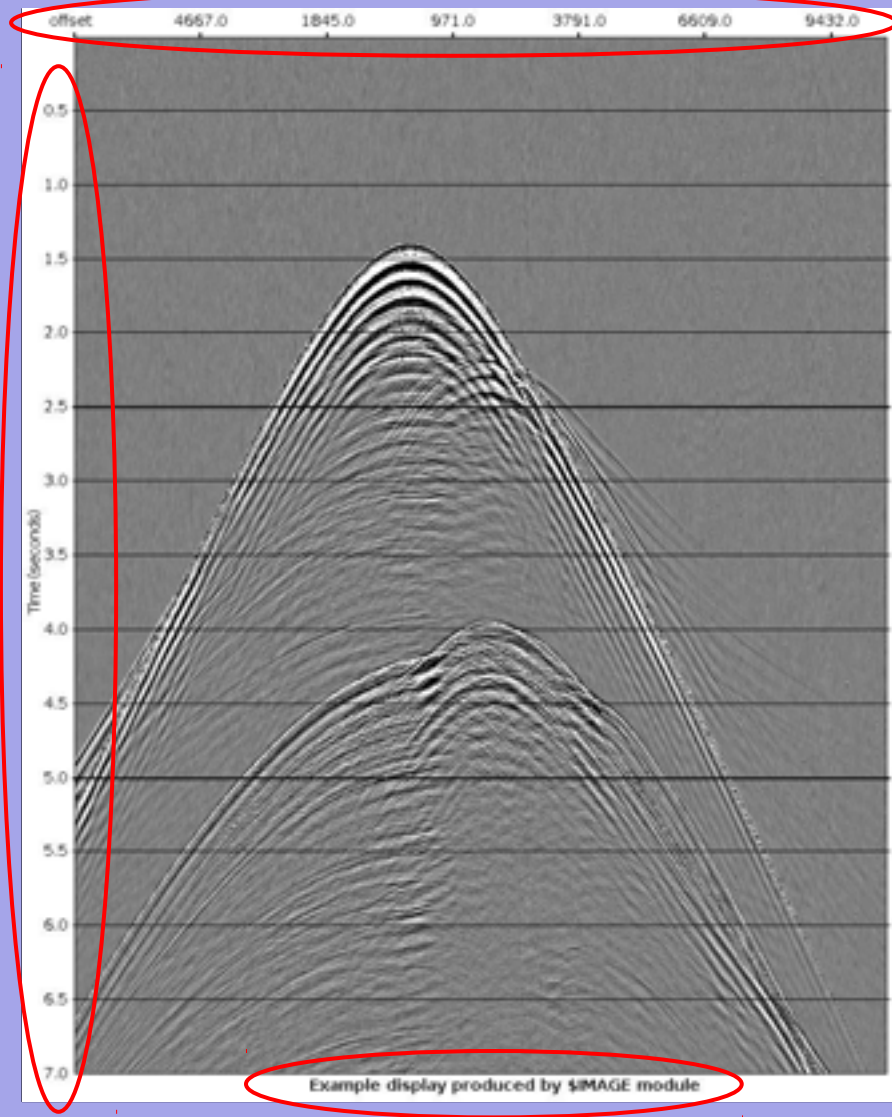
Screen dump created with **SeaView**

Image file created with **SeaSeis $IMAGE module**

Example display produced by $IMAGE module

SeaSeis – Future Plan

- No far-reaching real plan in place right now. Current plan is to do opportunistic updates and publish a new version every 6 months
- Still waiting to see if there is general interest, from users or developers
- ..maybe someone with complementary skills & interests in software development and geophysical coding may want to join me?
- Possible to wrap Seismic Unix modules in generic way (maybe also Madagascar..?). For example: Generic wrapper module for Matlab compiled code already exists (not published)
- If SeaSeis is discontinued, it would be nice if some of its concepts or tools were taken over by other systems

Demo