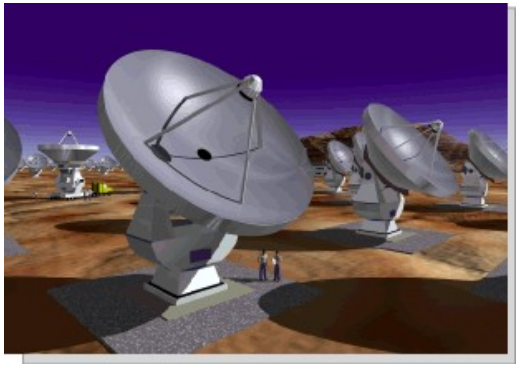


ACS Error System

Atacama
Large
Millimeter
Array

J. Avarias & A. Caproni

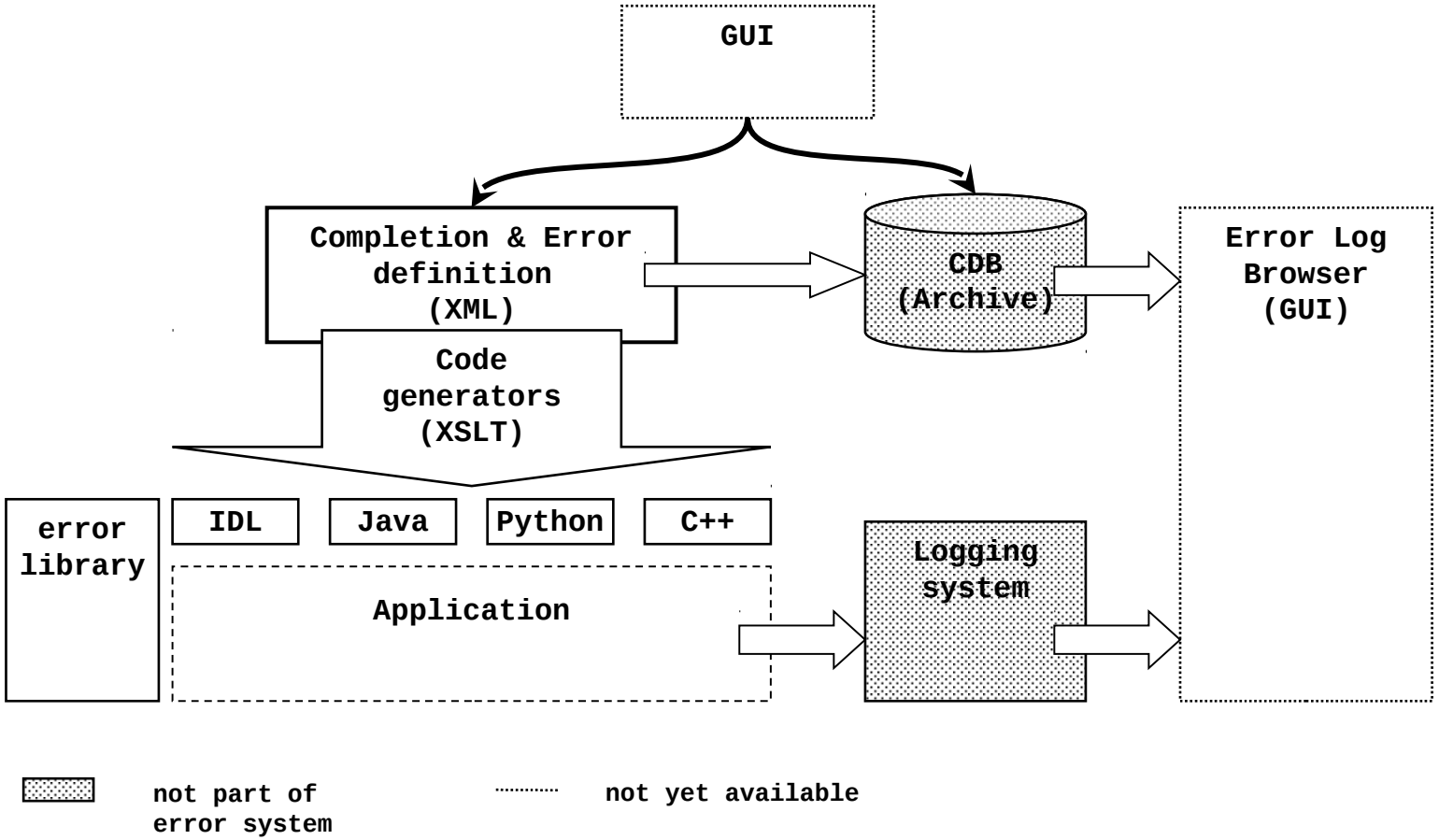
ALMA Project

Summary

- **Features**
- **Completions/Errors**
- **Architecture**
- **Error representation/structures/propagation**
- **Error Trace**
- **Code generation**
- **Defining new errors**
- **Programming language support**
- **Remote error handling**

ALMA Project

Architecture




ALMA Project

Error representation

- each completion/error is defined with two numbers:
  - type (group)
  - code
- OO mapping:
  - type -> namespace / IDL module
  - code -> completion class/ exception class

ALMA Project

Error structures

- **error trace** (can not exist standalone)
  - type+code
  - severity: Error, Critical, Alert, Emergency
  - timestamp
  - source code info: line, file, routine
  - run-time info: process name, thread ID, host name
  - arbitrary data in name-value format (generic) or members
- **exception classes** (contain error trace)
- **completion classes**:
  - non-error
  - error (contain error trace)

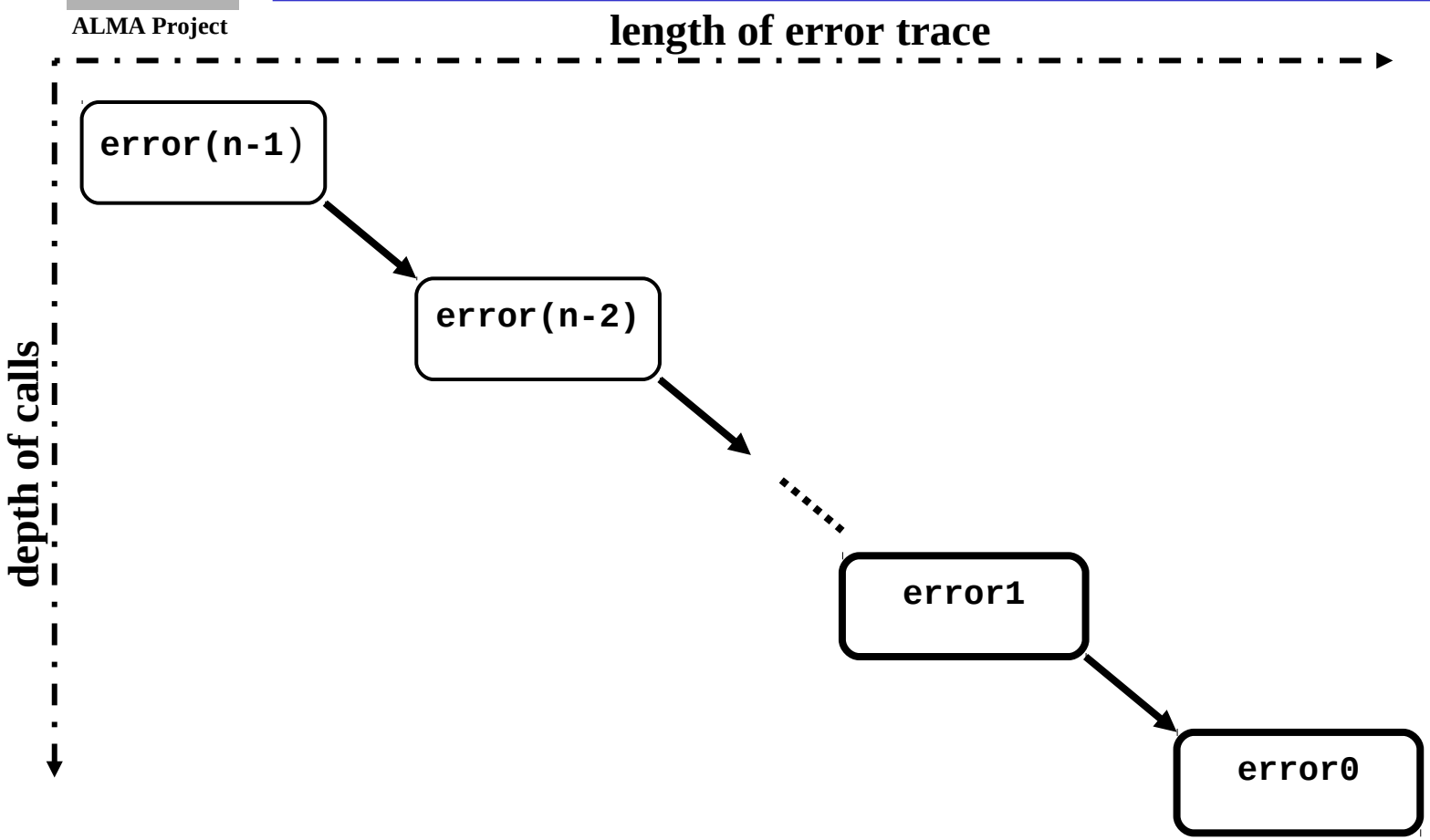

ALMA Project

Arbitrary data

- generic way: name-value pairs
- members:
  - defined together with error: name, type and description (documentation)
  - generated getter/setter methods
  - stored in name-value pairs (accessed in a generic way)

ALMA Project

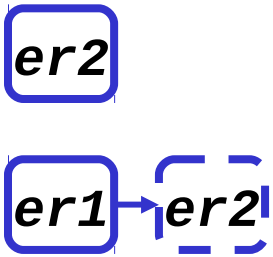
Error Trace



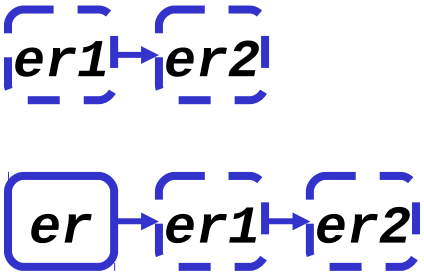
ALMA Project

Example of error trace

```
ACSErr::Completion* Ex::op(){
    er2 = new
      RecordNotFoundCompletion("getRecord");

    er1 = new
      PositionNotObtainedCompletion(er2,"getP
      osition")
    er1->returnCompletion();
}
```

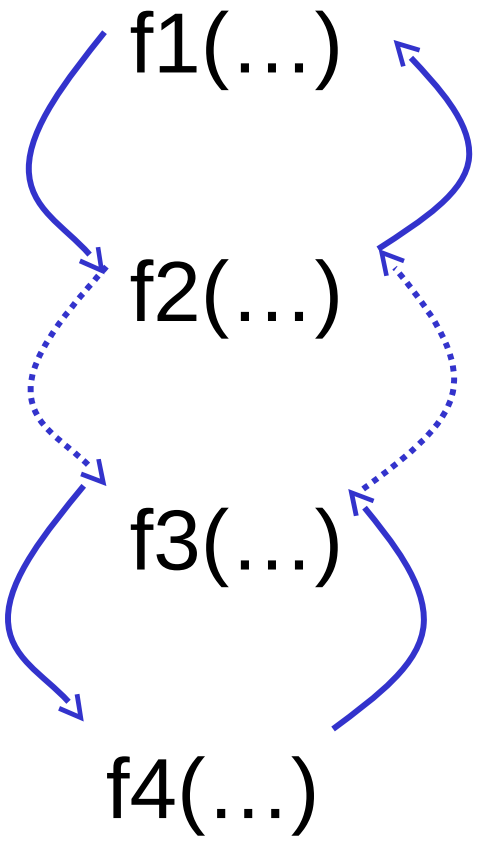


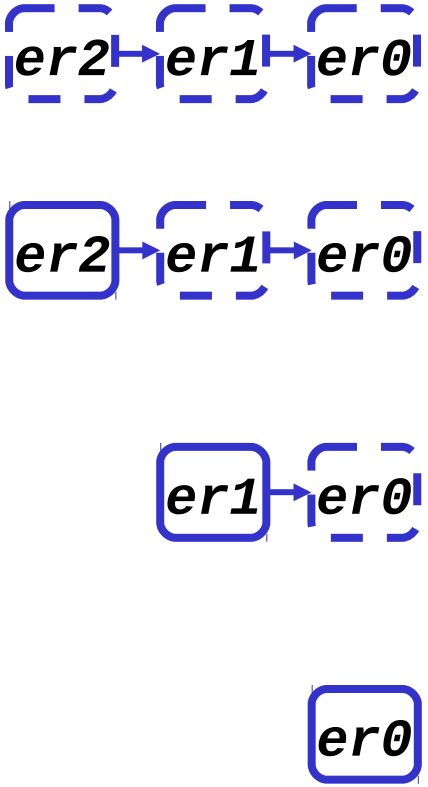

```
ACSErr::Completion *c = …->op();

   er = new PointingFailureCompletion(c,
   "routine");
   er->log();
```



ALMA Project

Example of Error Trace






ALMA Project

Error propagation

- throwing exception:
  - just for reporting errors
- returning/sending completion structure:
  - asynchronous actions/communications (callback mechanism)
  - reporting different success statuses of the operation
  - where exceptions are not supported e.g. old C++ compilers

ALMA Project

Error Definition (XML)

```
<Type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="ACSError.xsd"
   name="ExmplErrType“  type="900902" _prefix="alma">

<Code name="PointingInProgress" shortDescription="Pointing in
   progress"  description="Pointing in progress"/>

<ErrorCode name="PointingFailure" shortDescription="Pointing
   Failure“ description="Pointing can not be executed">
   <Member name="Azimuth"  type="double“
    description="Azimuth"/>
   <Member name="Elevation" type="double“
    description="Elevation"/>
</ErrorCode>

<ErrorCode name="PositionNotObtained"
   shortDescription="Position not obtained" description="Position
   of the object can not be obtained"/>
...
</Type>
```

ALMA Project

IDL: Error Trace (1/2)

- **IDL structure to hold error trace information: *`ACSErr::ErrorTrace`***
  - error: type & code , severity (Error, Critical, Alert, Emergency)
  - timestamp
  - runtime information: process, thread, host
  - source information: file name, line number, routine name
  - additional info: pairs of name-value
  - **“pointer” to the previous error**
- **Error trace is not used standalone, but as a part of:**
  - completion structure
  - exceptions

ALMA Project

IDL: Error Trace (2/2)

```
struct ErrorTrace {
   string file;
   long lineNum;
   string routine;
   string host;
   string process;
   string thread;
   TimeStamp timeStamp;
   string shortDescription;
   ACSErr::ACSErrType errorType;
   ACSErr::ErrorCode errorCode;
   ACSErr::Severity severity;
   NameValueSeq data;
   sequence<ErrorTrace, 1> previousError;
 };
```


ALMA Project

References

- http://www.eso.org/projects/alma/develop/acs/OnlineDocs/ACS_Error_System.pdf