Data analysis software generator, ROME & ARGUS

R. Sawada, S. Ritt, M. Shneebeli
Mar, 30, 2006
The Physical Society of Japan
61st Annual Meeting
Ehime University
Introduction
ROME was developed in MEG collaboration in 2004.

ROME is independent of experiments. It is a general purpose tool.

Several groups are using it in U.S., Canada and Europe.

ROME works under Linux, Windows and Mac. (Maybe under other UNIX like system with minor modification.)

ROME is neither an analysis software nor a library. ROME generates analysis software from an experiment definition XML file.

Generated software is written in C++.

Generated software works on the basis of ROOT library.
Concept

- **Automatic** code generation
  - **Less work** of physicists. (Physicists must spend time for physics, instead of programing!)
  - **Less bugs** than hand written code
  - Better **readability** of code.
  - Always **compatible document** with code.

- Separation of calculation and data container classes (**tasks** and **folders**)
  - Flexible control of calculation. For instance, you can try different re-construction algorithms without re-compiling.

- Sharing experiment independent part with other groups
  - **Improvements** from other experiments reflect on **your analysis**. (vice versa)
Analyzer is basically a command line program which reads input files and write result in output files.

It allows you to go into interactive session.

It runs also in batch mode.

You can make GUI which shows result of analysis. (ARGUS is GUI part of ROME)
Analyzer is basically a command line program which reads input files and write result in output files.

- It allows you to go into interactive session.
- It runs also in batch mode.
- You can make GUI which shows result of analysis. (ARGUS is GUI part of ROME)
Code generation
How does ROME work?

**ROME Project**

- MEGAnalyzer.xml
  - Experiment definition XML file

**ROME**

- romebuilder.exe
- Base classes
- Utility classes

[http://midas.psi.ch/rome](http://midas.psi.ch/rome)
How does ROME work?

ROME Project

Experiment definition XML file

MEGAnalyzer xml

romebuilder.exe

Base classes Utility classes
How does ROME work?

Project

- Experiment definition XML file
- MEGAnalyzer.xml

Generate

- Pre-defined class files
- MEGAnalyzer.html
- Makefile
- .h .c .cpp files

ROME

- romebuilder.exe
- Base classes
- Utility classes

http://midas.psi.ch/rome
How does ROME work?

**ROME Project**

1. **Experiment definition XML file**
   - MEGAnalyzer.xml

2. **Pre-defined class files**
   - MEGAnalyzer.html
   - Makefile
   - Pre-defined class files

3. **Compile**
   - meganalyzer.exe

4. **Generate**
   - romebuilder.exe
   - .h files
   - .cpp files
   - Base classes
   - Utility classes

[http://midas.psi.ch/rome](http://midas.psi.ch/rome)
How does ROME work?

Project

User

Edit

MEGAnalyzer.xml

Experiment definition XML file

Edit

Pre-defined class files

Compile

meganalyzer.exe

Generate

rome

Base classes
Utility classes
Folder definition

Experiment definition file

```xml
<Folder>
  <FolderName>XEPMT</FolderName>
  <FolderDescription>Data for each PMT.</FolderDescription>
  <Field>
    <FieldName>QE</FieldName>
    <FieldType>float</FieldType>
    <FieldComment>Quantum efficiency</FieldComment>
  </Field>
  <Field>
    <FieldName>Gain</FieldName>
    <FieldType>float</FieldType>
    <FieldComment>PMT gain</FieldComment>
  </Field>
</Folder>
```

Always same naming convention of methods to access folders.

```
gAnalyzer->GetXEPMT()->GetGain()
```

gAnalyser Manual

**XEPMT**
Data for each PMT.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QE</td>
<td>float</td>
<td>Quantum efficiency</td>
</tr>
<tr>
<td>Gain</td>
<td>float</td>
<td>PMT gain</td>
</tr>
</tbody>
</table>

**CMPMTData**
Folder Class for Calorimeter Photo Multiplier data.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Task definition

Experiment definition file

```
<Task>
  <TaskName>XEEnergyRec</TaskName>
  <TaskDescription>Reconstruct energy of gamma.</TaskDescription>
</Task>
```

- **updates framework**
- **generates task class**
- **updates document**

User implements actual calculation into generated skeleton.

```
void MEGTXEEnergyRec::Init()
{
}

void MEGTXEEnergyRec::BeginOfRun()
{
}

void MEGTXEEnergyRec::Event()
{
}

void MEGTXEEnergyRec::EndOfRun()
{
}

void MEGTXEEnergyRec::Terminate()
{
}
```
Auxiliary
Run time configuration

Configuration file allows you to change analyzer’s behavior without re-linking.

- **Online/offline?**
- **Input data format.**
- **Batch mode or not.**
- **Database connection**
- **Controlling tasks**
- **Controlling trees. (I/O)**
- **User defined steering parameters**
ROME has flexibility which allows user to customize software.

- User editable **Makefile**
- User defined **I/O class** for user’s input files
- User defined **database class**
- **Tasks and folders are editable** by users. (ex. adding data members or methods)
- Linking **user defined classes**
- **Steering parameters** in configuration file. **Command line options**
SQL database is getting popular to store calibration constants and so on.

ROME has built-in functionality to access SQL database.

ROME supports MySQL, PostgreSQL, SQLite and more.

You need not learn SQL programming. We introduced path system to specify variables.

Example: Reading gain of 10 PMTs for run number 100.

```
Read(pContainer, "/RunCatalog[id=100]/Calibration(0,9)/gain");
```

- `pointer to container to hold values`
- `path`
Any ROOT object can be published over network. This feature is useful for online monitor. GUI display is also a ROME project.
ROME is developed as a general purpose software generation tool for elementary particle physics. (Actually, it is currently used not only HEP)

ROME takes care experiment independent part. What users have to do is basically implementing calculation. It allows you to make analysis software like writing macros.
ROME is a framework generator for event based data analysis. It has been developed for the MEG Experiment at PSI Switzerland. In the ROME environment, the experimenter defines the analysis framework for his experiment in a very clear and compact way in an XML file. Out of this framework definition file, ROME generates all experiment specific classes of the framework. The experimenter only needs to add the calculation code to pre-generated event methods. Contact person: Matthias Schneebeli or Stefan Ritt.

ARGUS: is a framework generator for graphical display of experiments. It has been developed for the MEG Experiment at PSI Switzerland. It obtains histograms from MIDAS analyzer or ROME, and obtains data from SQL, XML, ODB database and ROME. ARGUS was designed to be a general purpose framework for online monitor of high energy physics experiment. Once user defines tabs and folders in an XML definition file, ARGUS builder generates prototype of source codes which already have main window creation, connection to histogram/database server and update function of folders. User can design own application with putting graphical objects such as canvases, buttons, sliders and so on... Contact person: Ryu Sawada or Stefan Ritt.

ROODY: is an application based solely on ROOT for histogram display. It is meant for display of .root files and .hbook files. If running this application in conjunction with either the Midas analyzer or the ROME analyzer, it will allow to visualize online data retrieved through a socket port. Contact person: Joe Chuma or Stefan Ritt.