GEM Computing Subsystem Meeting - SSCL

August 4, 1993

Abstract:

Agenda and presentations of the GEM Computing Subsystem Meeting held at the SSC Laboratory on August 4, 1993.
Agenda for Computing Subsystem Meeting

9:00-12:00 Wednesday 4/Aug/93 in Directorate #2:

Introduction (KMF) 10'
Some GEM computing projects (IS) 15'
Project Methodology/Management (KMF) 20'
Project Template (JH) 15'
Test Beam Computing Proposal (GW) 20'
break 10'
L2 timing studies (HU) 15'
GISMO (TB) 30'
Geometry Wkshp Conclusions (JW) 15'
Where are we going next? (All) 30'
M. Atiya (BNL)
John Branson (UCSD)
Toby Burnett (U. of Wa.)
Laird Cornell
B. Dalesio (LANL)
Peter Dingus
Milind Diwan
Jim Dunlea (FNAL)
Yuri Fissiak
Vladimir Glebov
John Hilgart
Jean Huang
Tony Johnson (BU/SLAC)
Tom Kozlowski
Hong Ma
Gena Mitselmakher
Antonio Morelos
Richard Mount (CalTech)
Ken McFarlane
S. McKee
Roger McNeil
Harvey Newman
Frank Paige
Lee Roberts (SSCL)
Gary Sanders
Irwin Sheer
Michael Shupe (U. of Arizona)
Rick Shypit
Carl Timmer
Henk Uijterwaal
Torre Weanus (LLNL)
Bill Wisniewski
John Womersley
Gary Word
Jin Chu Wu
George Yost
S. Youssef (FSU)
Overview

Objective

Have working computing systems now......1995-??
for GEM - design phase, test beams, operation 1999? 2002?...

Plan

1. Analyze overall problem
2. Subdivide into projects
3. Allocate manpower to projects.

How to implement and manage the plan?
GEM TDR has a preliminary analysis of the processes. One can identify a list of projects from that analysis without completing the analysis (see later discussion by I.S.)

Examples

Interprocess Communication
Data Description Language

Geometry Description

It is to complete the list of projects including top-level project, e.g., Analyze system.

Project management is an issue; there are different approaches.
Deliverables are defined at each stage.
Boehm Spiral Model (modified)

SURVEY
 Alternatives
 Risks
 Dependencies (requirements)

REVIEW, COMMIT
 Goals, resources
 constraints, schedule
 Plans

RISK REDUCTION
 Evaluate
 Risk resolution

PROTOTYPES...

PLANNING

Investment by turn

Stage 0: (charter) 1 eg 2 man-days
1: 10 eg 3 man weeks
2: 100 eg 8 man-months

Phase: angle
Stage: Turn no.
This is an O-O Development model; the O-O aspect appears in the iterative part, where the analysis is done in O-O fashion.
The application of the spiral model to GEM system development could be in the form of making one turn of the spiral for each stage of a "traditional" (IEEE, GEM TDR, PRCD) life cycle, with deliverables corresponding to each stage.

A set of linked documents would then evolve as the spiral is traced out. Hypeertext could be used to provide the links. Template for "spiral turn document" exists.

Comments

G.W. Hard to apply spiral model to projects underway

T. Kork: Same comment; used IEEE model and found it helpful

Maybe the "spiral" approach is a management technique, not a project plan document system?
List of Computing Subsystem Projects

I. Sheer, G. Word, J. Womersley

- Framework Projects

- Detailed GEM Detector Simulation and Reconstruction projects

- Test Beam (TB) specific projects
Framework Projects

- Job Control Parameters
  Timescale: 5/93 -- 7/93
  People: I. Sheer, G. Word

- Software Technology Choices
  Timescale: 5/93 -- 9/93 (some choices final) to 3/94 (all choices final)
  People: Framework Group, T. Wenaus, (+ SDC and PRCD)

- Data Modelling and Module Definition
  Timescale: 9/93 -- 6/94
  People: All groups
  Prerequisite: IDL, DDL
Detailed GEM Detector Simulation and Reconstruction projects

- Geometry Description Language and Tool
  Timescale: 6/93 -- 12/93
  People: J. Womersley, T. Burnett, M. Seman, (+ interest from SDC)

- Describe GEM using the Geometry Description Tool
  Timescale: 3/94 -- 9/94
  Prerequisite: Geometry Description Language and Tool
  People: J. Womersley, H. Uijterwaal, + each subsystem

- Describe 1995 Test Beam setup using the Geometry Description Tool
  Timescale: 1/94 -- 6/94
  Prerequisite: Geometry Description Language and Tool
  People: J. Womersley, H. Uijterwaal, G. Word, + each subsystem

- Monte Carlo Generators and interfaces, including simulation of event pileup
  Timescale: 5/93 -- 7/93
  People: I. Sheer, G. Word
• Particle Transporter through the GEM detector
  Timescale: ongoing -- 3/94
  People: T. Burnett, G. Word, J. Womersley, H. Uijterwaal

• Digitization
  Timescale: 3/94 -- 12/94
  People: H. Uijterwaal, Detector Subsystem Groups (Mu: T. Wenaus)
  Prerequisite: Describe Test Beam setup using the Geometry Description Tool; Data Modelling and Module Definition

• 3D Event Display
  Timescale: 10/93 -- 6/94
  People: T. Burnett, G. Word, T. Wenaus
  Prerequisite: Selection of GUI technology; Geometry Description

• Full GEM Reconstruction
  Timescale: 3/94 -- 12/94 (at least Test Beam part)
  People: Detector Subsystem Group (Muon: T. Wenaus; Cal: ?; Tracker: ?)
  Prerequisite: Geometry Description; Digitization
Test Beam (TB) specific projects

- TB Global Control System
  Timescale: 6/93 -- 12/94
  People: V. Glebov, H. Uijterwaal

- TB Data Acquisition / TB Online
  Timescale: 6/93 -- 12/94
  People: A. Morelos, J. Branson, J. Dunlea, H. Uijterwaal, T. Weanus

- TB Reconstruction
  People: Detector Subsystem Group (Mu: T. Weanus)
  Corequisite: Full GEM Reconstruction

- TB Database(s)
  Timescale: 1/94 -- 6/94
  People: G. Word, K. McFarlane, + PRCD

- Calibration Studies
  Timescale: 1/94 -- 9/94
  People: Hong Ma, (Mu: T. Weanus)
  Prerequisite: Digitizations
• Mass Storage
  People: J. Hilgart, G. Word, L. Cormell
To:       R. McFarlane  
From:  I. Sheer, J. Womersley, G. Word  
Re:     List of Computing Subsystem Projects  
Date: May 28, 1993

There are two main strands to the Computing Subsystem effort, namely the detector simulation and event reconstruction effort and the effort to evolve all of the computer software and hardware systems that will be required for the final GEM detector. The evolution effort will use the GEM Test Beam at Fermilab as its next step.

Below we list some of the Computing Subsystem projects which support these two efforts. We include an estimate of the timescales of the projects which is driven by the expected start date of the Test Beam run at Fermilab which is 3/95. Note that THE CHOICES MADE BY THE INDICATED TIMESCALES ARE TO BE VIEWED AS WORKING DECISION FOR THE TEST BEAM 1995 EFFORT and NOT binding on, for example, the expected Test Beam effort in 1997 that may occur at the SSC Laboratory. We also include some of the people who are expected to work on the projects.

The Detector Subsystem Group mentioned below is lead by S. McKeen for the Tracker, J. Womersley for the Calorimeter, and R. McNeil and M. Atiya for the Muon subsystem. The Framework Group is lead by I. Sheer and G. Word. The global Physics Group is lead by J. Womersley.

1) Framework Projects (coordinated by I. Sheer and G. Word)

1) Job Control Parameters
   Timescale: 5/93 -- 7/93
   People: I. Sheer, G. Word
   Prerequisite:  
   Description: Most programs need to have several parameters set at run time to control the operations of the program during job run. Examples are the number of events to generate, the file to use which describes the detector configuration, etc. A prototype tool exists, which is written in C++. It remains to be determined whether the prototype is sufficiently flexible to satisfy the requirements of the task.

2) Software Technology Choices
   Timescale: 5/93 -- 9/93 (some choices final) to 3/94 (all choices final)
   People: Framework Group, T. Wenaus, (+ SDC and PRCD)
   Prerequisite:  
   Description: There is a long list of technology choices to be made so as to minimize the number of software packages that are required in the GEM environment. It is also important to attempt to choose the same technology as SDC, where feasible. This effort is being given high priority, since the technology choices will affect all software development. The areas in which a technology will be chosen include:  
   - RPC (inter-process communication), RPC (remote procedure calls), IDL (interface definition language), DDL (data description language), GUI (graphical user interface), CASE tools (computer-aided software engineering tools) and data base(s).

3) Data Modelling and Module Definition
   Timescale: 9/93 -- 6/94
   People: All groups
   Prerequisite: Interface Definition Language; Data Description Language
   Description: After the IDL and DDL are chosen, an integrated effort will commence to describe the input and output data of all of the software modules.

II) Detailed GEM Detector Simulation and Reconstruction projects (coordinated by J. Womersley)

1) Geometry Description Language and Tool
   Timescale: 6/93 -- 12/93
   Prerequisite: Geometry Description Language and Tool
   People: J. Womersley, T. Burnett, M. Seman, (+ interest from SDC)
   Description: Design a text-file language that can be used to describe a detector: build a tool to parse the language; provide the capability of making the appropriate calls to GEANT and other GEANT-like packages, such as the SLAC GISMO package, to instantiate the detector: eventually provide a method of modifying the detector description using graphical or menu-driven interfaces.

2) Describe GEM using the Geometry Description Tool
   Timescale: 3/94 -- 9/94
   Prerequisite: Geometry Description Language and Tool
   People: J. Womersley, H. Uijterwaal, + each subsystem

3) Describe 1995 Test Beam setup using the Geometry Description Tool
   Timescale: 1/94 -- 5/94
   Prerequisite: Geometry Description Language and Tool
   People: J. Womersley, H. Uijterwaal, G. Word, + each subsystem

4) Monte Carlo Generators and interfaces, including simulation of event pileup
   Timescale: 5/93 -- 7/93
   People: I. Sheer, G. Word
   Prerequisite:  
   Description: Provide a means by which Monte Carlo event generators can be used to simulate SDC interactions and produce generator-independent representations of the simulated interactions; provide a mechanism by which the interactions can be combined into an event and similarly events into an event with pileup. The current project is to rewrite the GEMgen package in the C++ language and to switch from using Zebra to using other methods of data transport, such as Sun's xdr package.

5) Particle Transporter through the GEM detector
   Timescale: ongoing -- 3/94
   People: T. Burnett, G. Word, J. Womersley, H. Uijterwaal
   Prerequisite:  
   Description: While GEANT is the current particle transporter used by GEM, this project is expected to evaluate whether other possibilities, such as the C++ GISMO application that is being developed by a SLAC-based group, could be used for the 1995 Test Beam effort as well.

6) Digitisation
   Timescale: 3/94 -- 12/94
   People: H. Uijterwaal, Detector Subsystem Groups (M.U. T. Wenaus)
   Prerequisite: Describe Test Beam setup using the Geometry Description Tool
**Tool: Data Modelling and Module Definition**

**Description:** Each subsystem will provide routines that simulate the digitized responses of their subsystem to SSC events.

7) **3D Event Display**

**Timescale:** 10/93 -- 6/94
**People:** T. Burnett, G. Word, T. Wenaus
**Prerequisite:** Selection of GUI technology; Geometry Description

**Description:** Provide a 3D visualization tool that permits real-time rotation of the detector shell, detector digitizations and particle traversal through the detector.

8) **Full GEM Reconstruction**

**Timescale:** 3/94 -- 12/94 (at least Test Beam part)
**People:** Detector Subsystem Group (Muon: T. Wenaus; Cal?: Tracker?)
**Prerequisite:** Geometry Description; Digitization

**Description:** Each of the subsystems will provide code that interprets the detector digitizations and produces event hypotheses. The Detector Subsystems Group and the Global Physics Group will provide the code which integrates the subsystems code into global event hypotheses.

**III) Test Beam (TB) specific projects (coordinated by G. Word):**

1) **TB Global Control System**

**Timescale:** 6/93 -- 12/94
**People:** V. Glebov, H. Uijterwaal
**Prerequisite:**

**Description:** The global control of the partial-detector in the Test Beam will mainly be accomplished through the use of the EPICS system.

2) **TB Data Acquisition / TB Online**

**Timescale:** 6/93 -- 12/94
**People:** A. Morelos, J. Branson, J. Dunlea, H. Uijterwaal, T. Wenaus
**Prerequisite:**

**Description:**

3) **TB Reconstruction**

**Timescale:** 6/94 -- 12/94
**People:** Detector Subsystem Group (Mu: T. Wenaus)
**Prerequisite:**

**Corequisite:** Full GEM Reconstruction

**Description:** The reconstruction project for the Test Beam should re-use the same code as is used in the Full GEM Reconstruction, plus some additional code that will be required by the unique Test Beam setup.

4) **TB Databases**

**Timescale:** 1/94 -- 6/94
**People:** G. Word, K. McFarlane, + PRCD
**Prerequisite:**

**Description:** Choices will need to be made about which databases should be used for the Test Beam project. The number of databases is expected to be minimized. Special purpose databases are expected to be discouraged.

5) **Calibration Studies**

**Timescale:** 1/94 -- 9/94
**People:** Hong Ma, (Mu: T. Wenaus)
**Prerequisite:** Digitizations

**Description:** Studies will need to be performed on how the data collected at the Test Beam can be used for calibration purposes, especially for the calorimeter components.

6) **Mass Storage**

**Timescale:** 6/93 -- 6/94
**People:** J. Hilgart, G. Word, L. Cornell
**Prerequisite:**

**Description:** The means by which the Test Beam data will be stored on and accessed from tape will be provided by this project.
Proposal for the organization of GEM software projects
J.Hilgart, I.Sheer, G.Word

We have several requirements in mind when developing this proposal. These will be detailed later.

1) There will be two main areas (directory trunks) where GEM software may be located:
   - $GEM$ - for the publicly released sources, binaries and documentation.
   - $GEMDEV$ - for the developers.

2) The $GEM$ area will have this directory structure (a capitalized name is not a literal directory name).
   - $GEM$/src/ORG/PROJECT/VER[/PACKAGE][/SUB] - multiply-mounted afs-volume
     - /bld - "build" area for linked sources and binaries
     - /bin - full of links for non-version-critical utilities
     - /lib
     - /man
     - /doc
     - /etc - put templates here
     - /include/ORG/PROJECT/VER[/PACKAGE]/VER/PACKAGE
   - /bld/PROJCT[/PACKAGE]/VER/
   - /demo/ORG/PROJECT/VER - mandatory demo for this package
   - /example

3) We propose the following "ORG" directories:
   - gem/{gemfast,gemgen,gemutil,sim}
   - prd/{isajet,hippo,gismo,cheetah}
   - cern/{cernlib,geant}
   - lund
   - stdhep
   - gnu

4) For every project under the gem organization there will be a project leader, who will have write access to his project tree, as well as the $GEM$/bin area. He will be expected to make software releases for his project as necessary. User demand, developer demand, and the project leader's discretion should determine when it is necessary to provide a release. A GEM software librarian--currently JH--can be called on to help make the release.

5) A developers' area will be provided for each project for the purpose of maintaining a source-code repository and storing project documentation. This area will have the following structure:
The "pdoc" area is for project documentation and not generally world-readable. The "pdoc" area at the same level as the cvs directory will be automatically generated and updated from the repository by a program to be written.

The project leader can opt for different organization, i.e., a cvs repository at the level of each package, if there seems to be reason to do so.

6) We also prescribe how a project should be laid out. The repository only contains sources:

```
cvs/{Makefile, <other source files>}
```

When the repository is exported and makefile run, the following directories and files should be created:

```
cvs/test/$HOSTTYPE/<test suite>
/lib       volatile links of the form
/bin       -> ../$HOSTTYPE/bin
/include    - link directory containing subset of headers that are externally visible
```

In the lib area, we will put all the .o object files as well as the .a archived files. The library names will have the form:

```
libXYZ.a
libXYZ_debug.a
[libXYZ_profile.a]
```

7) Projects or packages relying on other projects or packages should follow these guidelines:

1) Link with stable, release versions of other project libraries, header files, etc.
2) Include directives for other packages header files should follow the "one-slash" rule, i.e., should look like:

```
# include <"XYZ/XYZ_HEADER.h">
```

This will greatly reduce the possibility of name-space collision of header files.

8) Migration from development to release area should be possible with the help of an install script, which performs these functions:
6) We also prescribe how a project should be laid out. The repository only contains sources:

\[ \text{cvs/ (Makefile, <other source files>)} \]

When the repository is exported and makefile run, the following directories and files should be created:

\[ \text{cvs/test/$_$HOSTTYPE/<test suite>} \]

\[ \begin{align*}
/\text{lib} &| | \text{ volatile links of the form} \\
/\text{bin} &| | \text{ } \rightarrow ../_$HOSTTYPE/bin
\end{align*} \]

\[ \begin{align*}
/\text{include} &| | \text{ link directory containing subset of headers that are externally visible}
\end{align*} \]

In the lib area, we will put all the .o object files as well as the .a archived files. The library names will have the form:

- \text{libXYZ.a}
- \text{libXYZ_debug.a}
- \text{libXYZ_profile.a}

7) Projects or packages relying on other projects or packages should follow these guidelines:

1) Link with stable, release versions of other project libraries, header files, etc.

2) Include directives for other packages header files should follow the "one-slash• rule, i.e., should look like:

\[ \text{#include } \langle \text{XYZ/XYZ_HEADER.h} \rangle \]

This will greatly reduce the possibility of name-space collision of header files.

8) Migration from development to release area should be possible with the help of an install script, which performs these functions:

a) exports the cvs repository to a new version directory
b) runs the Makefile to make binaries and object files
c) builds additional links— as relative links—which will be necessary, such as for the man pages, and the bin, lib, and include areas.

9) Makefiles.

Makefiles are heavily relied on to get things right in this scheme. Therefore templates will be provided in a template area, and consultation from a GEM software librarian will be freely offered.

The gem organization will have a common Makefile, "Makefile.common", where some reasonable defaults are to be set up. All Makefile writers are encouraged to use this common makefile for initial setup. The template will include Makefile.common by default.

Each project Makefile will have version numbers of the other packages which are relied upon hard-coded in order to provide a record from which to reconstruct a binary many years later.

An example Makefile, and generally how to use a particular package should be made available in the example directory.

Implementation of the above scheme.
a) exports the cvs repository to a new version directory
b) runs the Makefile to make binaries and object files
c) builds additional links—as relative links—which will be necessary, such as for the man pages, and the bin, lib, and include areas.

9) Makefiles
Makefiles are heavily relied on to get things right in this scheme. Therefore templates will be provided in a template area, and consultation from a GEM software librarian will be freely offered.

The gem organization will have a common Makefile, "Makefile.common", where some reasonable defaults are to be set up. All Makefile writers are encouraged to use this common makefile for initial setup. The template will include Makefile.common by default.

Each project Makefile will have version numbers of the other packages which are relied upon hard-coded in order to provide a record from which to reconstruct a binary many years later.

An example Makefile, and generally how to use a particular package should be made available in the example directory.

Implementation of the above scheme.

We will create a development area $GEMDEV, in /usr/ssc/gemdev

We can simultaneously migrate towards the implementation of the release area within the existing /gem area without serious conflicts.

Example.

$GEM = /gem
$GEMDEV = /usr/ssc/gemdev
ORG = gem
PROJECT = gemfast
VER = v204 <- v + 3-digit version is recommended
PACKAGES = {libfsi, libfxb, libtrf, usrfast, gemfast}
develop. area:
$GEMDEV/gem/gemfast/cvs/{pdoc, libfsi, libfxb, libtrf, usrfast, gemfast}
demo area:
$GEM/demo/gem/gemfast/v204/{fsiold, fsisel, fsispg, trfdemo}

man pages:
$GEM/man/cat5/gemfast.5 ->
$GEM/man/cat3/{libfsi.3,libfxb.3,libtrf.3}
We will create a development area $GEMDEV$, in /usr/ssc/gemdev

We can simultaneously migrate towards the implementation of the release area within the existing /gem area without serious conflicts.

Example.

```
$GEM = /gem
$GEMDEV = /usr/ssc/gemdev
ORG = gem
PROJECT = gemfast
VER = v204  <- v + 3-digit version is recommended
PACKAGES = {libfsi, libfxb, libtrf, usrfast, gemfast}
devlp. area:
   $GEMDEV/gem/gemfast/cvs/(pdoc, libfsi, libfxb, libtrf, usrfast, gemfast)
demo area:
   $GEM/demo/gem/gemfast/v204/{fsiold, fsisel, fsispg, trfdemo}
```
GEM Test Beam Computing Progress Report

Gary Word
SSCL

presented at the
GEM Computing Subsystem Meeting
August 4, 1993
1) TB Global Control System
Timescale: 6/93 -- 12/94
People: T. Kozlowski, C. Timmer, V. Glebov, H. Uijterwaal
Prerequisite:
Description: The control and monitoring of the partial-detector (18 degree sector) and detector prototypes in the Test Beam will be accomplished through a Global Control System, which will make use of the EPICS system where appropriate. Graphical interfaces will be established for the GCS. Shift personnel will use GCS to monitor and control the Test Beam run execution and to take calibration data.

Activities and Documents:
93/ 7/ 6 Workshop, initial pass at requirements and schedules; included discussion of capabilities and limitations of the EPICS package. (GEM TN-93-434; for agenda see Article: 36 of ssc.gem.comp)
93/ 7/28 Rough Draft of System Requirements Specifications
93/ 7/28 Rough Draft of System Project Management Plan

2) TB Data Acquisition / TB Online
Timescale: 6/93 -- 12/94
Prerequisite:
Corequisite: TB Global Control System
Description: Provide front-end and event-building electronics and software to extract and assemble data from the Test Beam detector and store the data in a long-term data storage facility.

Activities and Documents:
93/ 6/16-17 Workshop, began the process of defining data sizes and rates, among other things. (GEM TN-93-418)
93/ 7/ 1 Workshop, considered DAQ requirements and possible solutions.
93/ 7/ 8- 9 Workshop, decided on leaders of the various TB computing tasks and on the general DAQ approach. (see Article: 19 of ssc.gem.calor for a meeting short summary or Article: 8 of ssc.gem.testbeam for a longer write-up).
93/ 7/12 Proposal for a switch based event-builder for the gem final testbeam (M. Botlo, M. Bowden, H. Uijterwaal) (see Article: 9 of ssc.gem.testbeam)
93/ 7/23 Near-term budget proposed by J. Dunlea (private e-mail)
93/ 8/ 3 Draft MOU between PRD Elec. Dept and GEM Elec/DAQ. (D. Trainer)

3) TB Reconstruction
People: Detector Subsystem Group (Mu: T. Wenaus)
Prerequisite: Test Beam setup using the Geometry Description Tool
Corequisite: Full GEM Reconstruction
Description: The reconstruction project for the Test Beam should re-use the same code as is used in the Full GEM Reconstruction, plus some additional code that will be required by the unique Test Beam setup.

Activities and Documents:
None-to-date.
4) TB Data Analysis
   Timescale: 1/94 -- 12/94
   People: Detector Subsystem Group (Mu: T. Wenaus)
   Prerequisite: Data Analysis Toolkit and Environment
   Description: The Test Beam data will be analysed by extending the
                standard Data Analysis Toolkit and working in the
                standard environment.

   Activities and Documents: None-to-date.

5) TB Database(s)
   Timescale: 1/94 -- 6/94
   People: G. Word, K. McFarlane, + PRCD
   Prerequisite: Test Beam setup using the Geometry Description Tool
   Description: Choices will need to be made about which databases should
                be used for the Test Beam project. The number of databases
                is expected to be minimized. Special-purpose databases
                are expected to be discouraged.

   Activities and Documents: None-to-date.

6) Calibration Studies
   Timescale: 1/94 -- 9/94
   People: Hong Ma, Mu: T. Wenaus
   Prerequisite: Digitizations
   Description: Studies will need to be performed on how the data collected
               at the Test Beam can be used for calibration purposes,
               especially for the calorimeter components.

   Activities and Documents: An effort should be made to find out about current studies.

7) Mass Storage
   People: J. Hilgart, G. Word, L. Cormell
   Prerequisite: Comprehensive set of GEM TEST BEAM requirements.
   Description: The means by which the Test Beam data will be stored on
                and accessed from tape will be provided by this project.

   Activities and Documents:
   93/ 6/ 2 Memo from L. Cormell asking for cpu and data storage
           requirements (see p. 117 of GEM TN-93-418).
   93/ 7/ 7 Rough draft replying to Cormell's memo.
           Numerous iterations have been drafted.
   93/ soon Another draft which focuses solely on requirements with
           no mention of implementation possibilities and with
documented justification for total event sample size required
           by each subsystem.
   93/ 7/16? GEM Testbeam data requirements briefly outlined at a
           PASS Collaboration meeting. PASS is considering and will
           most likely offer to assist in the area of data storage.
Level 2 Algorithm timing studies. 
(First Results).

Henk Uijterwaal, SSCL, August 4, 1993.
GEM Level 2.

* Processor farm with 100 to 1000 500 SSCUP processors, processors are used for L2 and L3 algorithms.

* 10 to 100 kHz Level 1 trigger rate.

* 1 to 10 ms available for the L2 trigger algorithms.

Algorithms.

* 3 layers:

1) Use data of a single component to extract trigger primitives: e/γ, jet, Et, tracks, muon.

2) Combine data from 2 or 3 components: CT-track + e/γ-candidate.

3) Extract global physics quantities: Invariant mass of 2 objects.

* Studies in progress are:

1) e/γfinding, trigger for H0→γγ (and a few other processes). Relatively simple algorithm but a lot of data to process. (Every cell in the EM-CAL).

2) Jet-finding. More complex algorithm but less data to process (sums of EM- and HAC-CAL cells). Will be presented in a next talk.
e/γ-trigger.

* Algorithm:

1) Extract the data from the GEMFAST commons. 3 possibilities: all cells, only the cells in a cone around the L1-e/γ candidates, only the towers where L1 found a e/γ.

2) Check if the cells are indeed e/γ. Use ratio of energy in 3x3/5x5 cells.

3) Make a sorted list of possible e/γ, remove e/γ that have been counted twice.

4) Calculate invariant mass of all pairs. Plot the pair with the highest invariant mass.

* The algorithm uses lookup-tables wherever possible.

* The algorithm is written in plain Fortran 77.

* The algorithm finds the same photons as Level 1.

* Results:

1) Can be used to separate physics from 1 and 2 photon background events. Almost all H⁰ survive a cut at 50 GeV.

2) Reducing the amount of data available to L2 reduces both the efficiency and reconstructed invariant mass.
Level 1 (1+2 photon trigger)

Level 2 (1+2 photon trigger)

Level 1 (1+2 photon trigger)

Level 2 (1+2 photon trigger)

Level 1 (1+2 photon trigger)

Level 2 (1+2 photon trigger)
Timing.

* Added a few calls to a clock routine to the code. The routine returns the CPU-time used by a program. From the accuracy of this routine, it follows that the accuracy of the timing results is about 10 ms.

* Typical processing times are of the order of 75 ms for H₀ events on an HP in PDSF for events with a L1 1 or 2 photon trigger.

* This includes time spent in extracting the data from GEMFAST, this is the job of the EDD's not the L2 processors. Substracting the time spent in this part of the code gives 63 ms/event.

* A HP is a 35 SSCUP machine. On a 500 SSCUP machine, the algorithm will take about 4.5 ms.

* The performance is independent of the amount of data made available to L2. Most of the time is spent in checking if the energy deposit is from an electron or photon.
Optimising the code.

The first version used about 90 ms/event.

Reducing the number of loops over the data gained about 10 ms.

Storing intermediate results gained another 5 ms (less accesses to the lookup tables, at the expense of more memory usage).

Removing subroutine-calls did not help.
Conclusions.

1. This can be used as a trigger for $H^0 \rightarrow \gamma \gamma$.

2. The algorithm can be executed in the time available for a L2 algorithm.

Next:

1. Do the same for a jet trigger algorithm.

2. Test the performance of a “realistic setup”: e/\gamma and/or jet-finding depending on the L1 trigger bits.

3. Use CT and Muon data.
Gismo: An Object-Oriented Approach to Particle Transport and Detector Modeling

Toby Burnett
University of Washington

- Introduction: brief description, list of collaborators
- Why OOP?
- Simple application as an introduction to Gismo and to OOP
- Overview: a client-server view
  - the Medium class
  - Geometry classes
  - Graphics
  - Interactions in matter
- Examples of initial use
- Status, future
What is Gismo?

- unfortunate name for a program
- not a program at all, but a framework, or toolkit for building applications
- no claims for superior geometry, physics or speed: only possibility to combine best ideas

Who

Bill Atwood          SLAC
Alan Breakstone     University of Hawaii
David Britten       McGill University
Toby Burnett         University of Washington
Gary Word           SSCL

Gone, but not forgotten: Mick Storr, David Myers (CERN)
Scattering example

Description: 20 MeV e\(^-\) on 0.25 cm H\(_2\)O: want scattered energy, angle

![Diagram](image)

Considerations for an interface:

- What does the user need to understand
- How hard is it to specify his requirements
- Especially, how are the user hooks specified?

**EGS4** - AUSGAB, HOWFAR, (Mortran macros)

**GEANT** - GUKINE, GUSTEP
The main() for a Gismo application:

```cpp
void main()
{
    MyApp mother; // declare and create an application
    // (called 'mother' since it is
    // a "mother volume")

    mother.init(); // define beam, geometry, detectors...

    mother.run(); // generate and simulate the events
    mother.last(); // dump or display histograms
}
```

what is MyApp? Our own class that inherits from a SlabApp, a special application class that makes it easy to study planar layered geometries

```cpp
class MyApp : public SlabApp
{
    void init();
    void last();
};

void MyApp::init()
{
    addSlab(0.25, "H2O");
    addSlab(0.1, "vacuum", *new Spectrometer);

    setBeam("e-", Hep3Vector(0,0, pbeam ),
             Hep3Vector(0,0,-1) );
}

void MyApp::last(){ /* do output of histograms... */ }
```
The definition of Spectrometer: our private subclass of Gismo's Detector

```cpp
class Spectrometer : public Detector
{
    public:
        void score(const GParticle&, const Hep3Vector&, float, float);
};
```

score() is virtual -- a subtle, but important point

```cpp
Histogram energyHistogram("electron energy",17,20,0.05);
Histogram angleHistogram( "angles", 0,90,1);

void Spectrometer::score(const GParticle &p, const ThreeVec&, float, float)
{
    if( p.charge()==0 ) return;

    energyHistogram.Fill( p.energy() * 1000. );
    anglesHistogram.Fill( p.theta() * 180/M_PI );
}
```

Note the services provided by the GParticle: charge, energy, angle

**Summary:**

what were the user hooks in the OOP implementation?
A client-server view of Gismo

client

server

main()

Application

event generator

Field

Medium

Particle Transport

Decayer

Geometry

Material

Interactor

PDT
GEOMETRY BUILDER WORKING MEETING
(8/3/93)

16 attendees including 3 from accelerator

John Womersley
   Introduction, Overview, Priorities, Risks

Yuri Fisyak
   Title constants used in SIGEM (Zebra TZ)

Dale Orth
   Modeling of Accelerator (mechanical, shielding) + video

Toby Burnett
   GISMO objects: mediums, volumes, graphics

Laurie Waters
   SABRINA graphical interface to MCNP/LAHET

Saul Youssef
   Review of geometric part of simulation: system of
   primitives used in Glvify package
Geometry Builder

**Initial requirements**
Later developments

**EDIT**
- Interactive geometry creation on the screen with a GUI and mouse
- Modification and positioning of volumes interactively on the screen with GUI
- Point-and-click selection of volumes on the screen, with display of parameters, positioning, materials, media and rotations.
- Drag-and-repositioning with mouse
- Merge geometries from various sources (e.g. to add a CAD model of a support structure, input from an IGES file, to a previously created GEANT model)

**DISPLAY**
- Display in 2D, with slices and projections
- Display in 3D, rotatable, hidden line removal

**INTERNAL REPRESENTATION**

**EXTERNAL REPRESENTATION**
- Persistent objects?
- ASCII file describing the geometry in some well-defined, human readable format.

**FILTERS**

**NATIVE REPRESENTATIONS**
- GEANT geometry in a GSAVE RZ file and/or
- Fortran code (calls to GSVOLU and GSPPOS) list of parameters available as an ascii file, must include creation of materials, tracking media and rotation matrix calls.
  - with optimization of GEANT geometry tree
  - handling of magnetic fields in tracking media

**CLIENTS**
- GISMO
- CAD modeller (IGES file—compatible with AutoCAD, Intergraph & Pro-Engineer)
- CALOR89
- MCNP
- Event display, reconstruction, …
USEFUL DISCUSSIONS

- Do we need to define an ASCII geometry language? Can a more generic system (ADAMO, Cheetah) be used?

- Do we want to define our own “internal representation” (our own solid model) + filters to other representations, or
  Choose one existing representation (GISMO? SABRINA?) as our internal

- Need to think more about needs of clients other than particle transport — event display, reconstruction, etc.

- Question of making geometry persistent
NEXT MEETING: AUGUST 25-26, FERMILAB

- understand more about GISMO (All)
  → Atwood's television show next week

- understand more about SABRINA (J.W, B.M.)

- scheme for ASCII file based on Cheetah (G.W.)

- poll users — needs, desires