

Trigger Database Review

May 18, 2005

Dennis Box, Hal Evans, Rick Jesik, Jim Linnemann (chair), Terry Toole

“We’re from the review committee, and we’re here to help”

This review report contains an Executive Summary, Background, Items of Note from the Review, Conclusions drawn by the reviewers, Recommendations, Answers to Questions in the Charge, and Appendices with more detailed information.

Executive Summary

Elizabeth Gallas (EG) has worked long and hard on the trigger database. It already provides critical services, and is near a new release with a better user interface. Time estimates had not included integration testing, which turned out to be the major bottleneck. Give EG more help, and let her lead this release to convergence and declare victory. Have someone start entering IIb triggers, as the database is believed to be ready. Add someone to work on new functionality (starting by helping with testing the current release?) under EG’s direction while EG works on the trigger transformation tool. Offer EG an assistant/senior mentor to nail down specs on new functionality. Decide how to negotiate contents of future releases.

Background of the review:

The experiment is successfully running quite complex triggers with hundreds of selection criteria, and has been doing so for several years. It is a critical component for data taking. The trigger database allows us to configure the complex D0 trigger system in a reliable and controllable manner. The current trigger database also provides bookkeeping for collected data in such a way that it is easy to access for anyone doing an analysis.

However, a number of trigger capabilities are not being used because the matching trigger database software is not available. More hardware upgrades are coming, and needed to cope with higher luminosities. No new trigger database software has been released since 2002. Manpower joined this fall: 3 CD members working between 20% and 50% time. From a time estimate produced in September 2004, some 16 FTE-weeks of work have been coded, but testing is incomplete and no new software has been released.

The reviewers had access to a test database using the current release candidate. EG provided reasonably detailed project documentation. The reviewers did not in general investigate technical details of the time estimate, nor (with the exception of D. Box) individual elements of the project.

EG was not able to comment in detail on the review or the effort estimates, as support to triggermeisters entering version 14.0 of the trigger and continued release testing was felt to be of higher priority during the week nominally allotted for comments. This seems symptomatic of the conflicts between immediate demands of online trigger support and longer-term development in this project.

Notable items from the review:

No new capability in the trigger database is believed to be required for Run IIb trigger hardware.

The lack of the proposed release requires significant extra effort by both triggermeisters and Elizabeth Gallas (EG), further slowing development and testing.

Some xml is still generated by hand, and EG also does some specialized entry of information into the database. Again, these activities slow development of database software.

The xml generator is described by EG as not explicitly part of the Trigger Database itself, and modifications to xmlgen were not part of the schedule.

The use of client-server technology for data entry has slowed development significantly, according to both EG and Lee Lueking. Changes in server technology caused client changes. The entry client user interface is rather database-centric: a higher-level, more trigger-oriented view might have been desirable.

No update of the September 2004 schedule information was given to the reviewers.

Testing is a major bottleneck. Testing should be understood to also include iterations on the user interface and its detailed functionality. But there were no time estimates for testing in the original schedule. All integration testing seems to be carried out by EG alone. EG felt that the testing burden had a larger effect on completion date than the 20%-50% participation of CD personnel (their useful participation may have been limited by the testing turnaround time). EG has never worked on the database portion of the project full time; 2/3 of her effort has gone into parts of the overall project outside the database per se: user support, specialized entry, xml generation, etc.

The current release candidate includes some items with no time estimate in the September schedule.

Communication between developers and user community has been imperfect. Trigger management saw slow progress and thus gave up on some items they wanted, and felt the developers were defensive. Developers felt that priorities were constantly shifting, and that trigger manager's lack of detailed technical knowledge led to "specifications" too vague to actually implement.

Conclusions drawn by the reviewers:

The project has been "2 weeks away from release" or "80% complete" for a long time. The coding and unit testing has taken 1.5 times the estimate, which is perhaps not atypical of software project estimates. Integration testing time appears to be nearly half the coding and unit testing time, was not estimated, and was pursued part time by a single person. Thus, calendar time for the current release has been about twice what might have been hoped for in September. The coding effort had an estimate of FTE 16 weeks and received about 87% of an FTE of effort since Oct 15, so one might have hoped for 18 weeks with perfect estimation (the end of February). Instead this work took about 24 FTE weeks of coding and so far about 10 FTE weeks of integration testing over some 32 calendar weeks (Oct 1 to mid-May). See Appendix for more detail.

Recommendations:

Short Term:

- 1) The developers should release the present code shortly after trigger list V14.0 has been entered in the database, before the Vancouver collaboration meeting if possible.
- 2) The group should develop a testing strategy. Clearly distinguish the tests which must necessarily precede a release, those which must be performed by developers, and those that can most efficiently be tested only by real use. We encourage the developers to discuss to what extent this testing can be automated. The technology used in testing the client report interface may also be helpful for the client entry interface. This may require some effort by CD personnel. See the Appendix for additional comments.
- 3) A physicist (a run II triggermeister?) should be named immediately to begin coding Run IIb triggers (within current restrictions) by July 15. This will clarify whether there really are no new database issues associated with Run IIb.
- 4) A physicist should be added to the development effort, with emphasis on providing functionality for trigger list V15 and beyond (L2 Or-ing/Splitting, pseudoterms, or issues which arise in the attempt to code Run IIb triggers), while EG concentrates on shorter term needs and the trigger list transformation tool.

Future releases:

To go forward, one needs specifications, and estimates of effort required (for various ways of dividing the work among the trigger database, the xml generator, COOR, and lower level subsystems). This is necessarily iterative and crosses subsystem boundaries. Because EG has been a database developer, author of the xml generator, and an expert at inserting trigger information into the database, sometimes confusion has resulted and data entry problems were described as trigger database issues.

- 5) Nikos should chair a meeting at Vancouver to decide the specific options and facilities for which it worth making effort estimates. He should appoint someone to provide the specifications in sufficient detail for EG and others to use as a basis for the effort estimates. Some examples of facilities needing further specification are how to use the existing L2 software capabilities of going beyond 128 bits; and trigger exposure groups. See the Appendix for more detail.
- 6) The current schedule information is insufficient to support proper priority decisions. The review panel made a first draft of rough effort estimates based on available information (see the Appendix). Some items must be better specified before realistic estimates are possible. EG should correct the rough spreadsheet, including realistic estimates based on experience for integration testing and tuning of the user interface. This estimate will be imperfect, as new personnel may require learning time and new technology might help regression testing go faster.
- 7) It does not appear that the proposed future release schedule is fully matched to priorities. Release contents should be decided by the developers and Gavin or designee, taking into account the needs of the users (Trigger panel/ Trigger Commissioning for new capabilities; triggermeisters and physics analysts for user interface and developers project internal needs). Nikos and Gavin need to negotiate this split of responsibility, though it may become less problematic as future projects are less coupled; see Appendix.
- 8) After priority decisions have been made, the schedule information should be evolved to a release schedule with calendar dates. Future releases should be smaller in scope and more frequent.
- 9) For the painfulness of error correction during trigger list entry, consider whether a technical fix (such as exporting the current state of the database) might be easier than the full trigger list transform tool, or at least useful stopgap. The transformation tool seems a rather ambitious project for the current state of the project, but it may be the correct way to cope with increasing complexity. Possible technical fixes are discussed in the Appendix.

Responses to Review Questions

Technical/scope: is the existing design and associated implementation appropriate to satisfy the needs of D0? Are goals well defined? Agreed upon by all?

This scope is generally reasonable for the needs, though some functionality has already been de-scoped. The goals are agreed in principle but are sometimes imprecise, and progress has been slow. The technical choices have been taxing to the personnel available.

Schedule: can the schedule for implementation be met given the available person power and their expertise, and does its timescale meet the needs of the experiment? Assess level at which technical aspects are understood, planned and/or designed. Drill down into a few specific components.

Understand the scope of those components, the technical needs, how the schedule estimate for those components was determined, which resources are needed to meet that schedule.

The goals and resources are not matched. We did not extensively investigate design issues. There is no actual schedule available; rather there were estimates for effort required of CD personnel added to the project. This wasn't turned into a schedule with calendar dates. There was no specific estimate of effort required for integration testing and interface tuning, which has proved to be a major bottleneck: the effort is not negligible with respect to rest of the project, and has fallen mostly to one person. We have assumed that the current 1 FTE of CD personnel beyond EG will continue to be available.

The committee attempted to construct easier-to-digest schedule information in the form of a spreadsheet based on the information provided during the review. The spreadsheet lists the various projects, with FTE coding estimates, but not integration testing. EG spent about 1/3 of her time on this activity. We would now estimate testing as potentially-2/3 the coding estimates originally given. It also examines the coding/unit testing effort expended to date, perhaps 50% greater than the estimate of last fall (not uncommon in software projects). Finally, there is an attempt to quantify the "80 % of testing is done" statement by counting modules modified tested, and still to be tested. In the metric of components tested, 80% could well be accurate. EG has been asked to review these spreadsheets for accuracy. These cross checks, and discussions with the developers, led us to believe that a prompt release is a reasonable goal. However, investing another 20% of the testing effort to date would delay the release a full month: a release by mid-June implies a change of strategy. The spreadsheets form part of the Appendix.

Priorities: Is the current sequence in component development appropriate? If there are schedule concerns, is re-sequencing or partial de-scoping an option? Which components?

The current release grew in scope to include an update of server technology. This uncovered bugs in the client-server interface, and has delayed the release of present functionality. Not all future projects currently have estimates, so understanding their scope should be a high priority. It appears that many future projects are decoupled, so that prioritization (after the scope is understood) should be a productive exercise.

Resources: will the available resources allow a successful, on schedule completion of the project? If not, what is the type of expertise that could help? What would be the optimal way to integrate such expertise into the project?

More effort in physicist-level development and integration testing is clearly called for. This would most likely come in the form of someone reasonably familiar with the trigger. See Recommendations.

Management: is the management process, including interactions between D0 and CD adequate to deliver the product on schedule? This project requires interactions between people in different

management structures. Understand the path of communication and comment on its adequacy and possible improvements.

There are some problems here. Perhaps a senior advisor able to talk to all parties could help in turning high-level functional requirements into specifications for the project. Several possible projects need such definition in order to assess scope. This is a high-leverage activity which may not require a large effort.

The developers tend to use a narrower definition of the “trigger database project”: new code development explicitly interacting with the relational database. Users tend to view everything upstream of COOR, from data entry to xml generation, as the “trigger database”. This has led to much mutual misunderstanding, and even on the review committee, factor-of-two misunderstandings of where effort has gone. More of EG’s time goes to the “peripheral” parts of the project than to database code per se.

In addition, it appears that there are no bars to beginning implementation to a Run IIb trigger list: effort should be applied to the data entry. This does not appear to require new effort in the Trigger Database project. Someone familiar with the Run IIb trigger could do this.

A senior advisor might also help (for a limited period) with the definition of future releases, with input from trigger management on priority and the trigger database developers on the effort required and the dependencies. But the needs of the primary direct users, the triggermeisters, must also be represented in such decisions. Nor can the secondary users, physicists doing analysis, be neglected. They must be represented via the trigger panel or its representatives. The internal development requirements, the user interface problems, and new trigger functionality must be balanced in decisions on contents of new releases of code.

The committee did not look closely into the communication paths within the Computing Division, but there may be some issues there as well. Early in the project, a longer-term CD technical contact rather than a succession of consultants would have been beneficial. We hope that the CD personnel presently involved will be able to provide continuity going forward. Also, the SAM server and the Trigger Database server coexist sometimes uncomfortably on the same machine; maintenance outages for SAM are viewed as happening with little notice.

This project relies very heavily on a single person, as reflected in today’s agenda. There is an implicit risk if that person becomes unavailable. What would the best course of action be in that case?

The current project leader, EG, has been in charge for quite some time, but has worked with no other D0 physicist. It is in any case appropriate to begin sharing that burden (development, decisions, liaison with users, and integration testing) with another physicist. Testing and/or entering Run IIb data might be a natural way to begin. If the pace of development need continues to be high (for example, due to need for tuning the user interface), two people sharing the position on an ongoing basis might be appropriate for a time.

Appendices:

The appendices contain: Project Management comments, Further Background, Restated Priorities for New Functionality, What is L2 Or-ing and Splitting, Technical Options, Effort Estimates, Actual Effort, and Estimated Test Coverage.

Comments on Project Management

We all know but sometimes need reminding: the fundamental variables of a project are scope, schedule, and resources. You can (try to) specify any two, but the third is a logical consequence of those decisions. With fixed scope and insufficient resources, schedule suffers. We have asked for more resources for the project. Within that resource constraint, the scope decisions should be made in view of the likely schedule. Implementation is the choice of technical means and sequence within the given parameters, though different implementations may distribute the work differently and even change the overall effort required.

The scope includes functionality at 3 levels:

- New Trigger programming capability (to date the emphasis of the Nikos and the Trigger Board)
- User Interface: triggermeisters, physics analyzers, simulators, developers
- Internal: performance, security, maintenance...

Further Background on release plans

The current release has components with three purposes: allowing long term maintenance of the database server, automating time-consuming hand entry of items which EG now must enter for triggermeisters, and improvement of the user interface. Including the database changeover has delayed the release considerably.

The user interface has known limitations which the September plan addressed:

- More searching functionality is needed.
- Error correction is awkward.
- Considerably more manual entry is required than desirable.

The spreadsheets and the documentation provided give further project details. See

Agenda Server | Computing | Databases | Review | and the Trigger Database documentation http://www-d0.fnal.gov/~gallas/d0_private/trig/TDB_Review_0505.html .

Among the next proposed projects are:

Performance issues	4 weeks
new reporting	4 weeks
allowing L2preprocessor parameters to vary with trigger lists	7 weeks
enabling FPD triggers by allowing device group reporting	2-4 weeks
stopping hand entry of other items; better automated error correction	12 weeks
Enabling Level 2 branching or Or'ing	unknown
Enabling Level 1 pseudoterms	8-16 weeks
Version tracking between online code and trigger lists	4 weeks
Round Robin Streaming: xml generation only change	?
Scott Snyder and EG can specify this between them	

The times given here are coding and unit testing times. With current resources, calendar time could easily be twice as long for each project. The amount of integration testing may decrease if testing

automation works, or if future projects are narrower in scope. Until that happens, it is prudent to schedule for the present testing to coding time ratio.

The server is used for essentially all data entry now. Full exploitation of server technology has not been achieved with the personnel available. Part of the original motivation, data entry in support of simulation by general users, has been effectively de-scoped out of the project. Other purposes, such as isolation from particulars of database technology, may still have value (though the entire report interface avoids the server).

Restated priorities for new functionality from Iain/Trigger Board:

- 1) Integration of Run IIB trigger upgrade terms Needed for trigsim lists for V15 studies
- 2) Exposure Groups
 Needed for full usage of FPD; Marco sees this as a lower priority than 1)
 depends on relative priorities of experiment and integrating FPD into standard running
- 3) L2 Bit OR-ing not in V14 now, -> not the highest priority.

- 4) L1Cal Regional Terms and Dynamic Downloading
 L1 Cal Regional terms are complete and working (remove from list).
 Dynamic Downloading is not a trigger database issue (remove from list)

What is L2 Or-ing and L2 Splitting?

L2 Or-ing = multiple scripts for a single L1 bit, but detailed L2 reporting ignored by L3 and L3 scripts depending on the report summary for the L1 bit. A single L2 “superscript” is associated with each L1 bit. The effect is Or-ing several L2 scripts and running the same set of L3 scripts independent of which L2 script (for the L1 bit) passed. Best suited for groups of triggers attempting to accomplish the same goal. In principle, the present L3 code could do this, as L3 scripts depend on the 128 L1 bits as confirmed by L2, and that bit mask is the report summary which would be used for L2Oring: a L1 bit is confirmed if any of the L2 scripts of a L2 superscript succeed. Relationship is 1 L1 to many L2; 1 L1 to many L3

L2 Splitting = runs same L2 code, but detailed L2 results are used to select which L3 script to run. Retains the full model of a tree of specific trigger conditions for each L3 decision. Best when different processing in L3 is appropriate for each different L2 script passed. Has potential for more efficient L3 processing, since extraneous scripts not run, but will result in more L3 triggers to manage. One issue here is whether choice between Or-ing and Splitting is to be done for each L1 bit independently. Relationship is 1 L1 to many L2; 1 L2 to many L3

A meeting failed to generate clear specifications for TDB as to what was needed to implement L2 Or-ing. Depends on where in the system the L2 superscript becomes available. The L2 software produces both 128-bit (Or-ing) and 4096 bit (Splitting) result masks.

L1	L2	L3
A	B	one L2 script (B) associated to one L1 bit (A)
	C	present scheme: multiple L3 bits for a given (L2-confirmed) L1 bit
	D	both C and D run if B passes
L2 Or-ing		
E	F1	multiple L2 scripts (F1, F2, ...) associated with one L1 bit (E)
	F2	
	...	
	Fn	

	G	L3 script run if any of F1 or F2 or ... Fn pass
	H	L3 script run if any of F1 or F2 or ... Fn pass
	...	
L2 Splitting		
I	J1	multiple L2 scripts (J1, J2...) associated with one L1 bit
	K	Run if J1 passes
	L	Run if J1 passes
	...	
	J2	
	M	Run if J2 passes
	N	Run if J2 passes
	...	
...		

If you'd really wanted Or-ing, then $K = M$ and $L = N$ when expressed in terms of L2 splitting.

Likeliest scenarios:

- 1) Or-ing only, no planning for splitting (probably simplest);
- 2) Or-ing first, transition to splitting;
- 3) Both Or-ing and Splitting possible. Most complex: selecting L1 bit by L1 bit in a single trigger list: for example all three of these 3-level scripts (A, E, and I) in a single trigger list.

It must be decided what scenarios should be investigated; in any scenario there may be options for optimization by moving certain tasks among L2, L3, Coor, xml generator, and trigger database.

Technical Options for Automation of Entry Client Testing

The entry client web interface needs an automated test suite, and the path for implementation is straightforward. A good model for the testing is what Eric had done for the reporting interface. All the report web forms use the 'post' method to communicate the state of the textboxes, checkboxes, etc, which means that they all end up in the URL that communicates with the web server and then the Trigger Database. Eric has collected several hundred of these URLs, each representing some action on his reporting pages, and runs them through a script that compares the web server's reaction to what happened when the same input was given previously.

The client web interface uses the 'get' method to communicate its internal state with to the web server, meaning that its state is not put into the resulting URL. Changing all of these pages to the 'post' method is fairly simple. Elizabeth or some other expert could then cut and paste the URL's that result from manual client testing into a text file and they could be run automatically using Eric's script. Adding a new test to the suite means appending a URL to an input file and saving an html file to a directory.

Technical Options for Simplifying Error Recovery during Trigger List Entry

An 'export' of the trigger database takes only a couple of minutes. Doing this should be a prerequisite to entering a new trigger, at least until a better error recovery method is devised. There are also some possibilities for doing development on the development database and exporting a successful update to the production database. Each table has a "last touched date" column, so it is easy to recognize the changes and copy them out of a development database and into the real database after a successful change. The script below finds all trigger database rows modified by a given user in a given time frame and generates sql suitable for inserting these rows into another database.

Here's an example of using the script to copy out work elizabeth did on 2005-03-03

```
./find_changed.pl -S 'd0read/reader@d0ofprd1' -q "where MODIFY_USER = 'gallas' and  
MODIFY_DATE > '2005-03-02' and MODIFY_DATE < '2005-03-04'"
```

Here is the script

```
#!/usr/bin/env perl  
use strict;  
use DBI;  
use Getopt::Std;  
#####  
#find_changed.pl a tool for extracting all changed rows  
#in trigger database  
#author Dennis Box, dbox@fnal.gov  
#date Apr 2005  
#  
my $documentation = "usage: find_changed.pl options  
-S source connect_string  
    username/passwd\@db for Oracle  
    'username/passwd\@DBI:mysql:databasename;host=hostname' for mysql  
-q quaifier (where MODIFY_USER='fred' or something like that)  
-v verbose, use caution using this on big tables  
";  
#####  
my ($valid) = getopt('S:vq:');  
my $S_con_string='';  
$S_con_string = $Getopt::Std::opt_S if defined($Getopt::Std::opt_S);  
my ($user_S,$password_S,$dbname_S) = split(/\@/, $S_con_string);  
my $dbType_S='Oracle';  
$dbType_S='mysql' if ($S_con_string =~/mysql/);  
my $qualifier='';  
my $qualifier = $Getopt::Std::opt_q if defined($Getopt::Std::opt_q);  
my $verbose = $Getopt::Std::opt_v;  
my $last;  
my $col_types;  
my ($dbh_S) = DBI->connect($dbname_S,$user_S,$password_S, $dbType_S) ||  
die "Error connecting $DBI::errstr\n";  
if($dbType_S eq 'Oracle'){  
  &do_sql($dbh_S,"ALTER SESSION SET NLS_DATE_FORMAT='YYYY-MM-DD hh24:mi:ss'");  
}  
my @trigger_db_tables;  
@trigger_db_tables=&do_sql($dbh_S,  
    "select table_name from all_tables where owner='TRIGGER_OWNER'");  
my $table;  
my $col_names;  
my @cnt;  
my @data;  
foreach $table(@trigger_db_tables){  
  $table =~s/'//g;  
  #'  
  $col_names=&get_colnames($dbh_S,'select * from '.$table.' where 1=0');  
  if($col_names =~ /MODIFY_USER/){  
    @cnt=&do_sql($dbh_S,'select count(*) from '  
      .$table.' '.$qualifier);  
    if ($cnt[0]>0){  
      print "-- =====\n";  
      print "-- $table\n";  
      print "-- $col_names\n";  
      print "-- =====\n";  
      print "-- $cnt[0] rows found \n";  
    }  }}
```

```

    my @data=&do_sql($dbh_S,'select * from '.$table.' '.$qualifier);
    my $datum;
    foreach $datum(@data){
        print "insert into $table values( $datum );\n";
    }
}
}
$dbh_S->disconnect();
#
# return column names for a table
#-----
sub get_colnames
{
    my ($dbh,$cmd) = @_;
    $cmd =~ s/;//g;
    my ($handle) = $dbh->prepare($cmd);
    if ($DBI::err) {
        print "$DBI::errstr\n";
        print "col_names: error preparing sql: '$cmd'\n";
        return;
    }
    $handle->execute();
    if ($DBI::err) {
        print "$DBI::errstr\n";
        print "col_names: error executing sql: '$cmd'\n";
    }
    my $arr = $handle->{NAME};
    my $col_names = join(",",$arr[0..$#$arr]);
    $col_names;
}
#
# execute arbitrary sql, return results in an array
#-----
sub do_sql
{
    my ($dbh,$cmd) = @_;
    $cmd =~ s/;//g;
    my @processed;
    print("sub do_sql doing $cmd\n") if $ verbose;
    my ($handle) = $dbh->prepare($cmd);
    $handle->execute();
    if ($DBI::err) {
        print "$DBI::errstr\n";
        print "do_sql:error preparing sql: '$cmd'\n";
        #exit;
    }
    if ($DBI::err) {
        print "$DBI::errstr\n";
        print "do_sql error executing sql: '$cmd'\n";
        #exit;
    }
    my $arr2 = $handle->{TYPE};
    $col_types = join(",",$arr2[0..3]);
    my @results;
    while (@results = $handle->fetchrow) {
        if ($DBI::err) {
            print "$DBI::errstr\n";
            print "do_sql error fetching: '$cmd'\n";
            exit;
        }
    }
}

```

```

my($col);
my $i=0;
foreach $col(@results){
$col = '0' if $col eq '-0';
if ($$arr2[$i]==1 or $$arr2[$i]==-1
    or ($$arr2[$i]>=9 and $$arr2[$i]<13)){
    $col =~ s/'/'/g;
    $col = "".$col."";
    }
# $col = "".$col."" if ($$arr2[$i]==1 or $$arr2[$i]==-1
# or ($$arr2[$i]>=9 and $$arr2[$i]<13));
$col = 'NULL' if $col eq "";
$col = 'NULL' if $col eq '';
$i++;
}
my $rsult = join(' , ', @results);
print "$rsult\n" if $verbose;
foreach $col(@results) {
}
push(@processed,$rsult);
}
$handle->finish;
@processed;
}

```

The results of the above example are here.

```

-- =====
-- TL_MAPS
--
TL_NAME,TL_VERSION,TL_INDEX,CREATE_DATE,CREATE_USER,NAME,VERSION,EXG_NAME,EXG_VERSION,L3_TYP
E,NUM_NODES,MODIFY_DATE,MODIFY_USER
-- =====
-- 14 rows found
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 0 , '2005-03-03 22:04:14' , 'gallas' , 'SRTOOLS_ONLINE' , 5 , 'eg1' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 1 , '2005-03-03 22:04:14' , 'gallas' , 'min_bias_nim_NCU' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 2 , '2005-03-03 22:04:14' , 'gallas' , 'zero_bias_NCU' , 2 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 3 , '2005-03-03 22:04:14' , 'gallas' , 'L1CTT_DOWNLOAD' , 10 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 4 , '2005-03-03 22:04:14' , 'gallas' , '2CEM6_E15_SHT22' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 5 , '2005-03-03 22:04:14' , 'gallas' , '2CEM6_2E18_SHT22' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 6 , '2005-03-03 22:04:14' , 'gallas' , '2CEM6_E15_SH30' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 7 , '2005-03-03 22:04:14' , 'gallas' , '2CEM6_2E18_SH30' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 8 , '2005-03-03 22:04:14' , 'gallas' , 'E21_2L15_SH15' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 9 , '2005-03-03 22:04:14' , 'gallas' , 'E24_2L15_SH15' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 10 , '2005-03-03 22:04:14' , 'gallas' , 'T10_2E18_HARD' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 11 , '2005-03-03 22:04:14' , 'gallas' , 'T10_2E15X9_HARD' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 12 , '2005-03-03 22:04:14' , 'gallas' , 'T10_2E18_HARD2' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
insert into TL_MAPS values( 'test_L2_Oring' , 1 , 13 , '2005-03-03 22:04:14' , 'gallas' , 'T10_2E15X9_HARD2' , 1 , 'allcrates' , 1 ,
'regular' , 0 , '2005-03-03 22:36:00' , 'gallas' );
-- =====

```

```

-- TRIGGER_LISTS
--
TL_NAME,TL_VERSION,CREATE_DATE,CREATE_USER,STAT_USED,STAT_CURR,DAQ,TL_TYPE,TL_AUTOPAUSE,COM
ICS_RUNTYPE,HISTORY,MODIFY_DATE,MODIFY_USER,TL_OSTREAM
-- =====
-- 1 rows found

insert into TRIGGER_LISTS values( 'test_L2_Oring' , 1 , '2005-03-03 22:04:14' , 'gallas' , 'unused' , 'local' , 'primary' , 'physics'
, 'yes' , 'data' , 'Trigger list to create a template xml file which will be hand modified.
This list contains a set of triggers with conditions that will be used in the first test of L2 ORing. The xml file will be hand modified
in order to do the OR since a "super L2 script" concept does not exist in the trigger database application.' , '2005-03-03
22:36:00' , 'gallas' , NULL );

-- =====
-- TRIGGER_NAMES
--
NAME,VERSION,STAT_USED,STAT_CURR,CREATE_DATE,CREATE_USER,L1S_NAME,L1S_VERSION,L2S_NAME,L2S_V
ERSION,L3S_NAME,L3S_VERSION,L2_GLOBAL_CRATE,L2_GLOBAL_EXE,HISTORY,MODIFY_DATE,MODIFY_USER,STR
EAM_TYPE,TN_OSTREAM,L2_PASS_1_OF_N,L3_PASS_1_OF_N
-- =====
-- 4 rows found
insert into TRIGGER_NAMES values( '2CEM6_E15_SHT22' , 1 , 'used' , 'current' , '2005-03-03 15:17:10' , 'gallas' ,
'CEM(2,6)_ncu' , 1 , 'L2CALEM(15,x)' , 2 , 'mp17000_Ele(ELE_NLV_SHT,1,22.,0.,3.6,-99.,99.)' , 1 , 'global_crate' , 'global_exe' ,
'L1: Two calorimeter EM trigger towers with Et>6 GeV and not cal unsuppressed readout. L2: Requires a standard L2 EM
cluster with a threshold >= 15 GeV. L3: The trigger bit set to true if an electron is found satisfying tight shower shape
requirements with Et>22. GeV' , '2005-03-03 17:54:43' , 'gallas' , NULL , NULL , NULL , NULL );

insert into TRIGGER_NAMES values( '2CEM6_2E18_SH30' , 1 , 'used' , 'current' , '2005-03-03 15:22:34' , 'gallas' ,
'CEM(2,6)_ncu' , 1 , 'L2CALDIEM(18)' , 1 , 'mp17000_Ele(ELE_NLV_SH,1,30.,0.,3.6,-99.,99.)' , 1 , 'global_crate' , 'global_exe' ,
'L1: Two calorimeter EM trigger towers with Et>6 GeV and not cal unsuppressed readout. L2: requires the sum of the two
highest EM towers to be >= 18 GeV. L3: The trigger bit set to true if an electron is found satisfying loose shower shape
requirements with Et>30. GeV. Mark and Pass one in 17000 events.' , '2005-03-03 21:04:24' , 'gallas' , NULL , NULL , NULL ,
NULL );

insert into TRIGGER_NAMES values( 'T10_2E18_HARD' , 1 , 'used' , 'current' , '2005-03-03 16:14:15' , 'gallas' ,
'TTK(1,10.)_CEM(2,3)CEM(1,9)_ncu' , 1 , 'L2CALDIEM(18)' , 1 , 'L3FEle(ELE_NLV,2,15.,0.,3.6,-
99.,99.)_L3FEle(ELE_NLV_SH,1,15.,0.,3.6,-99.,99.)' , 1 , 'global_crate' , 'global_exe' , 'L1: Two calorimeter EM trigger towers
with Et>3 GeV. One of these towers must have Et>9 GeV. Also, the event must have one track with pt>10 GeV and NOT
Calorimeter unsuppressed readout. L2: requires the sum of the two highest EM towers to be >= 18 GeV. L3: The trigger bit set
to true if two electrons are found with Et>15 GeV satisfying loose requirements. One of the electrons also must satisfy loose
shower shape requirements.' , '2005-03-03 21:14:34' , 'gallas' , NULL , NULL , NULL , NULL );

insert into TRIGGER_NAMES values( 'T10_2E18_HARD2' , 1 , 'used' , 'current' , '2005-03-03 17:35:17' , 'gallas' ,
'TTK(1,10.)_CEM(2,3)CEM(1,9)_ncu' , 1 , 'L2CALDIEM(18)' , 1 , 'mp17000_Ele(ELE_NLV_SH,1,30.,0.,3.6,-99.,99.)' , 1 ,
'global_crate' , 'global_exe' , 'L1: Two calorimeter EM trigger towers with Et>3 GeV. One of these towers must have Et>9 GeV.
Also, the event must have one track with pt>10 GeV and NOT Calorimeter unsuppressed readout. L2: requires the sum of the
two highest EM towers to be >= 18 GeV. L3: The trigger bit set to true if an electron is found satisfying loose shower shape
requirements with Et>30. GeV. Mark and Pass one in 17000 events.' , '2005-03-03 21:48:13' , 'gallas' , NULL , NULL , NULL ,
NULL );

```

Depends

Purpose	payoff	Priorit:	On	Where	Parts	Modified	Task	wks	Who	Est	Wh	Est	Date	Rel	Date
Internal	long term maintenance	low		Server	client changes too, unfortunately		new ORB	2	sw		eg				
User Int	stop hand entry (EG)	high		Entry	status		bug fix	2	mv		eg				
User Int	stop hand entry (EG)	high		Entry	l1dialogs		rewrite	3	mv,sw		eg				
User Int	make less confusing	med		Entry	neoterms		rewrite	2	mv		eg				
User Int	cleanup & new reports	med		Report	objects, terms, L1dialogs, neoterms		rewrite	7	ew		eg				
-----end of proposed present cut-----															

Purpose	payoff	Priorit:	On	Where	Parts	Modified	Task	wks	Who	Est	Wh	Est	Date	Rel	Date
User Int	perf issues	med		Entry	obj,terms, scripts, names		rewrite	4	mv		4	eg			
User Int	cleanup & new reports	med		Report	scripts,tn,tl		rewrite	4	ew		4	eg			
User Int	allow ttransform			Entry	triggerlists		add functi	3	mv		3	eg			
User Int	allow ttransform			Report	tdependency		add functi	1	ew		1	eg			
User Int	stop hand entry, correction (Triggermeisters)	?high?	all user int'	Entry	ttransform		add functi	8	eg		8	eg			
								Total	12		12				
New capε	l2pp params by script	?low?		Entry	l2pp,terms,script		add functi	6	mv		6	eg			
New capε	l2pp params by script needed for ctt/stt??	?low?		Entry	xmlgen		add functi	1	eg		1	eg			
								Total	7		7				
New capε	crate lists	low		Entry	ed,dg		write	4	mv		4	eg			
New capε	crate lists	low		Report	ed,dg		write?	?	ew		?	eg			
								Grand To	31		31				
-----end of proposed next cut ?-----															

New capε	L1 Pseudoterms	?med?	l1dialogs	Entry,	report,	server	write	8-16	all		8-16	eg			
User Int	Versions: swe to Trigger list	?med?		Entry,	report,	DB schema	write	4	all		4	eg			
New capε	Run llb	high				no DB changes?		0							
New capε	Oring L2 bits	high		Allow	n L2	scripts per L1 script		?							
	Splitting L2 bits	?med?		?no	db	change?									
				new	connection	of L3 bits to L1/2 bits									
New capε	fpd (and ctt) triggering	?high?		Entry	neoterms,	triggerlists		1	eg		1	eg			
				Report	tl,	help_contents		1	eg		1	eg			
				Output	xmlgen			2	eg		2	eg			
trigdb means entry client															
tcb means report client															

32 wk so far

Effort compared to estimates

From Lee Lueking's records:
to May 15 actFTE estimated

Who started	wks	%	wks	wks/est	
mv Oct	28	50%	14	6	2.3 entry: much harder than anticipated (no python experience)
sw Nov	26	15%	3.8	3	1.3 server: harder than anticipated
ew Mar	10	25%	2.5	3	0.8 report: overall, about same as anticipated
vm Dec	8	50%	4	4	1.0 assumes Vijay Murthi on same projects as ew
	72		24.3	16	1.5 moderately harder than anticipated
			87% of an FTE averaged since October 15		

eg Sept 32 30% 9.6 1 9.6 much harder than anticipated (no testing estimate given)

First estimate from verbal rough recollections at TDB review

to May 15 actFTE estimated

Who started	wks	%	wks	wks/est	
mv Oct	30	50%	15	6	2.5
sw Nov	26	20%	5.2	3	1.7
ew Feb	14	25%	3.5	7	0.5
	70		23.7	16	1.5
eg Sept	32	70%	22.4	1	22.4

Modules and Testing: 1st cut, 2005

Modules and Integration testing

		Integration:			Integration:			
		modified	tested	untested	modified	tested	untested	
Entry	main							
Entry	aux				Report	aux		
Entry	roles							
Entry	status	y	y					
Entry	server	y	y					
Entry	objects		y?		Report	objects	y	
Entry	terms		y?		Report	terms	y	
Entry	scripts		y?		Report	scripts	y?	
Entry	triggernames			y	Report	tn		
Entry	triggerlists			y	Report	tl		
Entry	transform			y	Report	tldependency		
Entry	l1dialogs	y	y		Report	l1dialogs	y	
Entry	neoterms	y	y		Report	neoterms	y	
Entry	l2pp				Report	l2pp		
					Report	ed		
					Report	dg		
Output xmlgen		15	4	7	3	16	4	5
Subsystem:		15	4	7	3	16	4	5
Total:		31	8	12	3			0

about 1/4 of modules modified
 about half needed testing
 counting modules: consistent with 80% tested