Trigger Database Taskforce Status, Plans, Implications

Jim Linnemann
MSU
Trigger Board
February 6, 2006

TDB:

Conceptual trigger content

Entry (via server)

Reporting

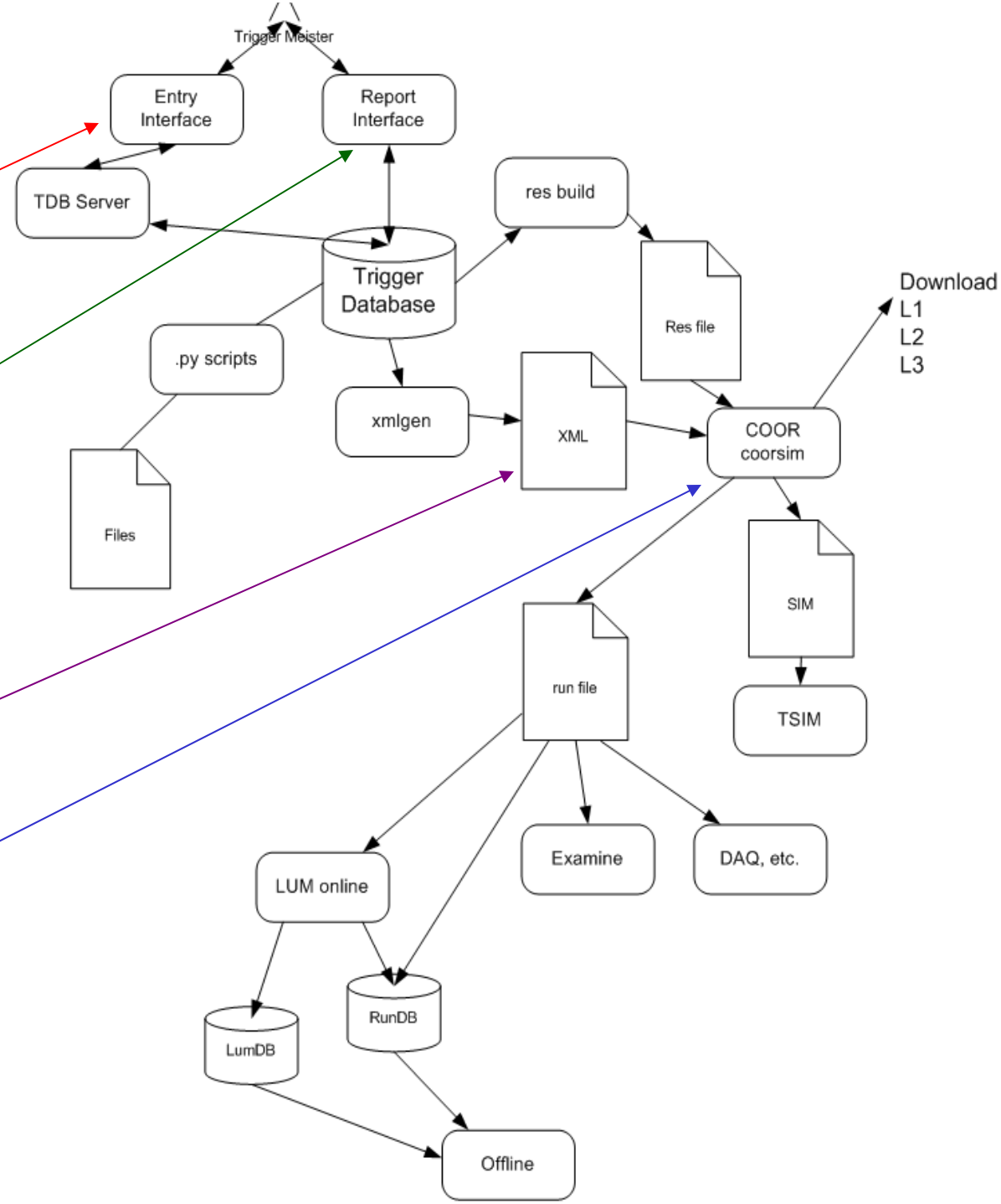
and, to Coor:

Xml output

Coor makes:

Bit assignments

Translation to Download



Trigger Database (TDB) Task Force Charge

Who: Co-heads: Igor Mandrichenko, Jim Linnemann

CD: Dennis Box, Margherita Vittone Wiersma, Stephen White

(differs somewhat from list in the charge)

D0: Arnold Pompos

Consultants: Elizabeth Gallas, Marco Verzocchi, Erik Wickland, Nikos Varelas (+ Scott Snyder, Michael Begel)

Start Nov 1 2005 (= old shutdown) Oct 28: first I'd heard from Igor

Charge: Consider/prioritize all requirements

by 30 Nov 8th draft then; final 12/5

Review all technologies

Make TDB maintainable

Implement by deadline **Feb 28, 2006 (= shutdown)**

working and tested product delivered

Weekly minutes to Amber, Gavin, D0 spokes

resource-loaded schedules neither requested nor produced

At Work | **Trigger** | Trigger Database Task Force

may need some clicking about certificates...

site started Nov 28, 2005 **linked to D0 Jan 15, 2006**

small text normal text large text

Plone

search

home news members

you are not logged in log in

you are here: home

navigation

- Home
- General Project Documentation
- Task Force Meeting Minutes

log in

Name

Password

log in

Forgot your password?

D0 Trigger DB Task Force

Taskforce Charge defines the structure, mission, scope and milestones for the Taskforce

List of requirements - what has to be done before March Tevatron shutdown and D0 upgrade

Details on Splitting/Orind

Taskforce weekly meeting minutes

Current Project status

Contact Us

Created by ivm
Last modified 2006-01-16 15:31

« February 2006 »

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

Detailed documents

Requirements Choices

Projects

Inclusive: be sure to ask for what we want

Consider items from old lists, but drop unless specifically decide to add

Some listed as “need better specification”

Many left to discretion of implementers: internal to TDB

Classify each project:

Time Scale Now, V14, **V15**, V15+ **(V15 = Feb 28)**

Priority **R**equired, **H**ighly **D**esirable, **D**esirable

Difficulty **S**traightforward, **M**oderate, **D**ifficult

(no DB change, add column/xml change, new schema)

Completion Requirements

C1) **T**rigger **T**ransformation **T**ool Req V15 Diff

remake list after replacing a Script Term (Meister)

have a detailed specification

implementation in progress, in “new technology”

first stage: substitution of L2/L3 Terms (no removal)

C2) Improved Error Reporting Desir

C3) Improved User Interface Desir

awaiting specification, and perhaps new tech

New Requirements

N1) Splitting and Oring

1a) xml generation Req, V15, Diff

1b) User Interface for large lists H.D., V15, Mod

N2) RunIIb tag support for L1Cal Req, Mod

N3) NoLum Desirable, Soon/Never, Mod

N4) L2Cal parameters Desirable/Mod/V15

xmlgen changes + ?

N5) Trigsim User Support H.Desir/Difficult

have specifications; needs new technology

Working Styles

CD prefers clean specification (by user only)

before discussing implementation

Physicists believe the project becomes better defined through working on it

we are used to being our own implementers

all requirements are negotiable if they are too costly!

iterative specification (this can drive implementer nuts)

no single expert (who signs off?)

This made requirements gathering “interesting”

some issues persist into detailed specification and testing but smoother…

Initial Projects

Provide useful (simple) functionality
Work way into system

while considering more complex functionality

Run2b qualifier for writing new L1cal terms
Done Jan 10

NoLum (save exposure groups; lum integrity)
Currently being tested by D0 (as of Jan 30)
xml has been generated
on 3rd iteration of D0 requirements
harder to define than to code?
Will need a test run with beam before shutdown

Technology Choices

We are keeping the Oracle TDB (no radical change)

TDB CORBA Server Replacement is under development:

- replace server with simulated one
- currently being coded
- can keep existing entry TDB code using the server

  - Enables other changes and maintenance
    - **Trigger Transformation Tool will use server replacement**
    - **New code much easier to develop**

TDB:

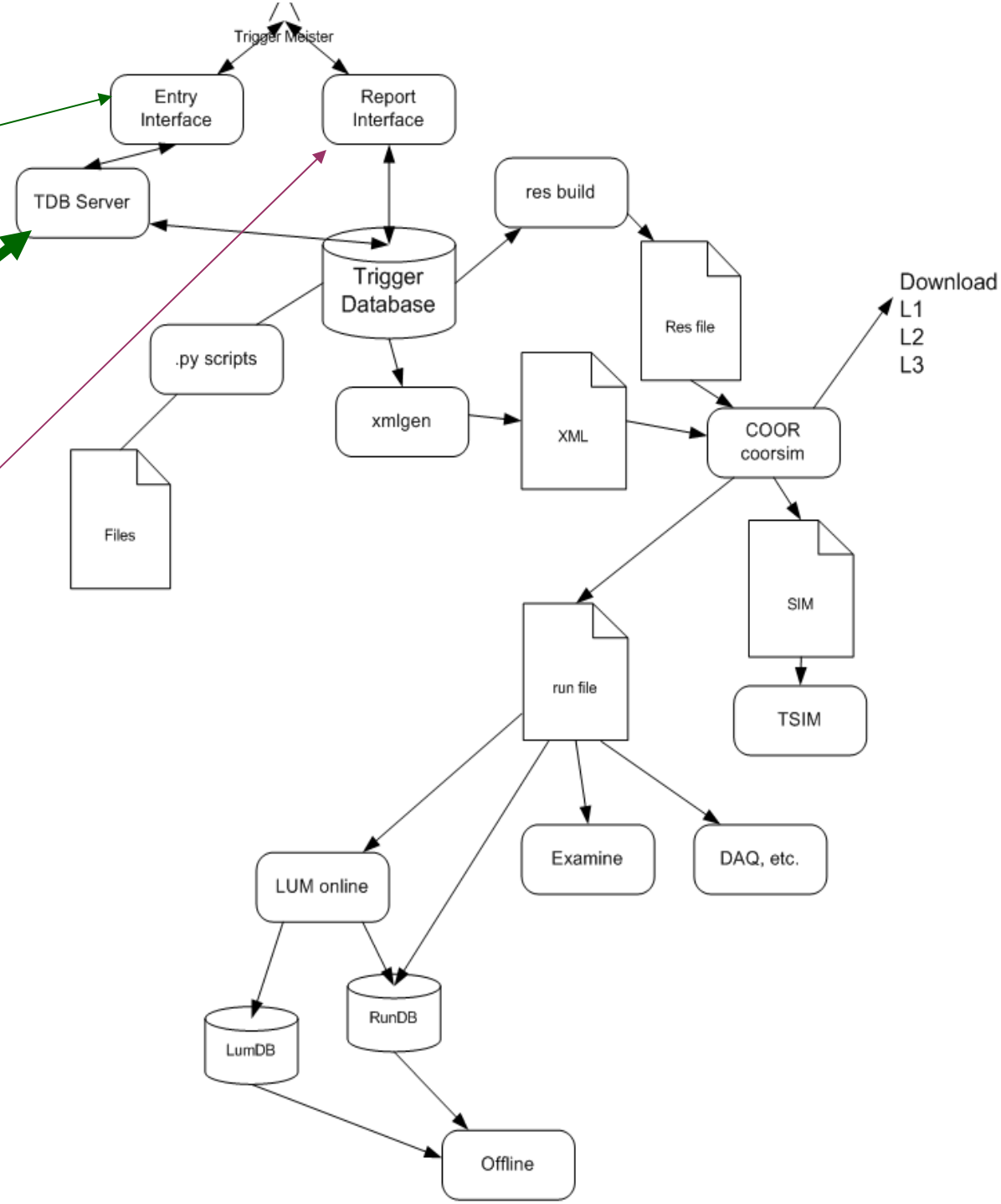
Entry

(via real or

simulated server)

Reporting

(no server)



Definitions: Branching, Splitting, Oring

L2 Branching: >1 L2 script/bit in L1 bit

Or of L2 scripts of L1 bit reported in 128 bit mask
Each specific L2 script reported in larger mask

Splitting: ≥ 1 L3 script for each such L2 script
(that is, if L2 branched)

Oring: ≥ 1 L3 scripts run if ANY L2 script
(i.e. the OR of the scripts) belonging to a L1 bit fire

Splitting and Oring:
Free L1 bits by using Branching in L2

Chose to offer both Split and Or

In a single trigger list

and even in a single L1 trigger

Partially since splitting simpler: stage if needed

Xml specs defined	Dec 12 (Marco, Scott et al)
Input Specs written	Dec 30 (Jim)
TDB implementation choices	Jan 3 (Jim, Marco, Igor)

Xml testing in progress

xmlgen new code draft	Jan 13
simple scripts entered in TDB	Jan 30

Boundary Conditions

Boundary conditions shape the solution

⇒ Splitting/Oring xml by Feb 28, 2006 !

Significant constraint on implementation in TDB

- Felt couldn't modify TDB structure on that time scale
- Found a way to implement within existing TDB

Problem was fit to technology rather than vice versa

Main casualty: no named group of Or'd L2 triggers

Oring Representation in TDB

Try L3 scripts y1 and y2 if any of L2 A-E pass

alas, no name for this L2 Or group, nor for the y1-y2 L3 Or group

Index	TrName	L1	L2	L3
1	L2A	L1	A	*Or
2	L2B	L1	B	*Or
3	L2C	L1	C	*Or
4	L2D	L1	D	*Or
5	L2E	L1	E	*Or
6	**L1ORY1**	L1	*Or	**y1**
7	**L1ORY2**	L1	*Or	**y2**

Use special values to alert xmlgen to structure

All *Or special values vanish before they get to xml

TriggerName fully meaningful only on L3 definitions

User interface issues:

Count L2 bits

Index is larger than # L3 bits

A test Oring script has been entered in TDB:

index	Trigger Name	Level 1	Level 2	Level 3
0	SRTOOLS_ONLINE / 5	This trigger definition includes a set of tools required by Level 3 ScriptRunner (a run configuration, an error handling tool and a geometry tool). Because it includes 'null' scripts at Level 1 and 2, it is not part of any specific trigger (a bit is not assigned), rather, it defines tools used by general programming instructions to Level 3 for this configuration to be listed before any trigger specific tools or filters in the element. This version has the error handling tool at the 'error' threshold.		
				SRtools_online / 5
The following triggers belong to the same Exposure Group. They share Device Group = allcrates / 1 and Exposure related L1 And/Or Terms: [ALiveBX & NOT(ASkip0) & NOT(Acaltc00)]				
1(1)	min_bias_nim_NCU / 1	requires beam crossing and N/S luminosity monitors above threshold in coincidence and NOT unsuppressed Calorimeter read out. This is the same as min_bias_NCU with the nim to denote run I electronics.		
		Afastz_ncu / 1	none / 1	pf1 / 1
2(2)	zero_bias_NCU / 2	requires beam crossing (an accelerator condition) and NOT unsuppressed Calorimeter read out		
		ALiveBX_ncu / 1	none / 1	pf1 / 1
3(3)	ARNOLD_OR1 / 1	L1 : E13_ISHT22 / 2 L2 : E13_ISHT22 / 2 L3 : placeholder		
		TTK(1,10.)_CEM(2,3) CEM(1,9)_ncu / 1	L2CALEM(15,x) / 2	L3_OR_MARKER / 1
4(4)	ARNOLD_OR2 / 1	L1 : E13_ISHT22 / 2 L2 : E14_2L15_SH15L20 / 2 L3 : placeholder		
			L2CALDIEM(18) / 1	L3_OR_MARKER / 1
5(5)	ARNOLD_OR3 / 1	L1 : E13_ISHT22 / 2 L2 : placeholder L3 : E13_ISHT22 / 2		
			L2_OR_MARKER / 1	mp17000_Ele(ELE_NLV_SHT,1,22.,0.,3.6)_Ele (IsoEle_SHT,1,22.,0.,3.6) / 2
6	ARNOLD_OR4 / 1	L1 : E13_ISHT22 / 2 L2 : placeholder L3 : E14_2L15_SH15L20 / 2		
				L3FEle(ELE_NLV,2,15.,0.,3.6)_L3FEle (ELE_NLV_SH,1,15.,0.,3.6)_L3FEle (ELE_NLV,1,20.,0.,3.6) / 2

User Interface Improvements for Large Splitting / Oring Lists

We are discussing *ideas* now

Entry interface enhancements
Assessing feasibility of:
- cut/paste/modify groups
- recognize groups automatically / modify group

Report interface enhancements
Assessing feasibility of:
- count L2 bits separately (my view: a requirement)
- represent split/or visually
- recognize groups automatically

Splitting and Oring
Effects on Online, Offline Software

Intrinsic: Oring has no single L1*L2*L3 condition

L2 scripts/bits no longer 1:1 with L1 bits

OR of all L2 scripts still 1:1 with L1

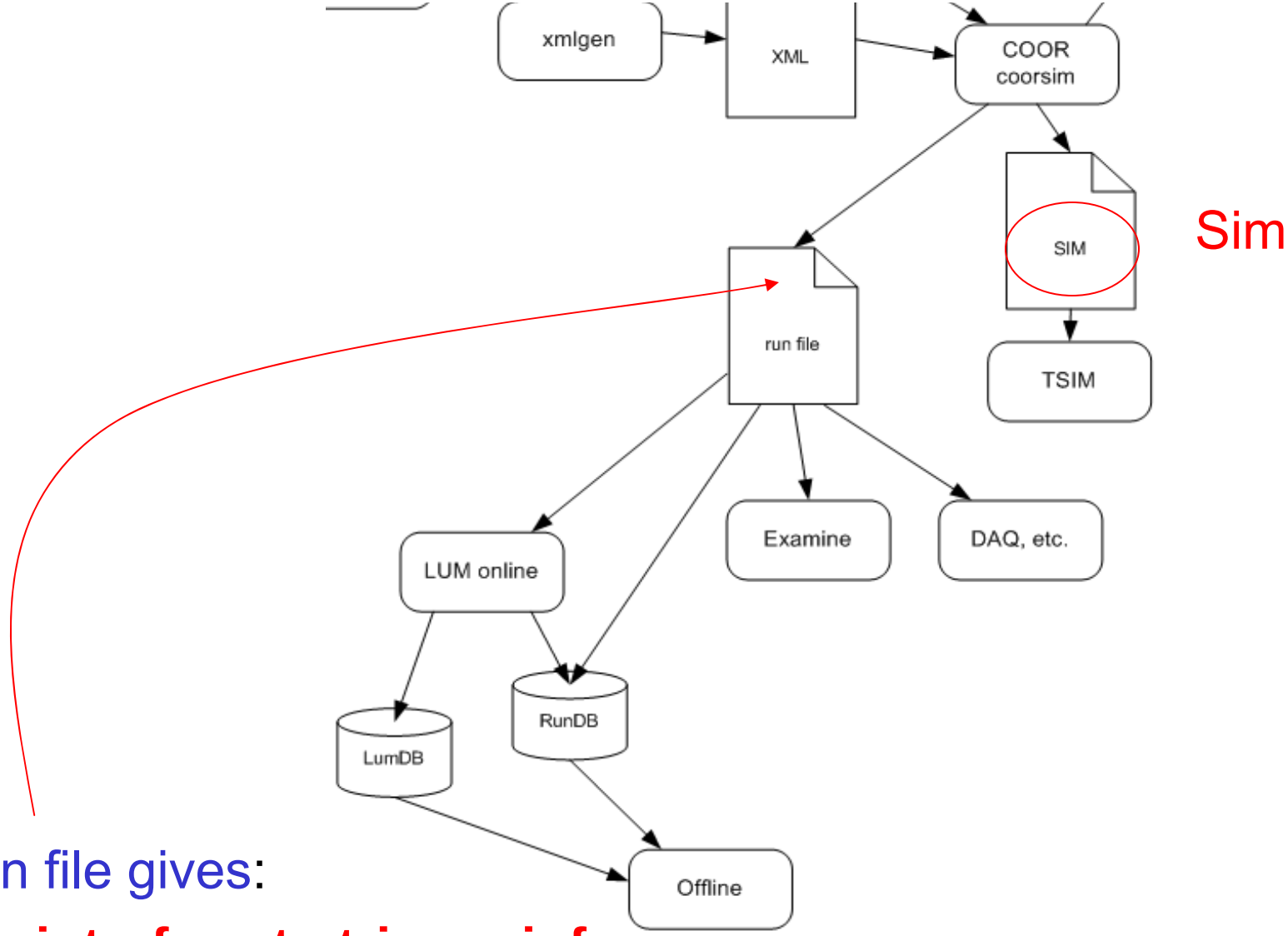
Still: one real TriggerName for each L3

COOR/Runs DB associates all bits to names used in TMB/CAF and all online

Specific: from Oring choices in TDB

No new issues **UNLESS** read TDB directly

some Python scripts do so



Coor run file gives:

Primary interface to trigger info

Bit assignments:

Trigger Names to bits

L1: L2: L3 bits

Serves this info to online

And via RunsDB to offline

Coor Run File

Generated from hand-written xml

L2bit: **2 2** L2OR-l1bit3
L2bit: **3 3** L2OR-l1bit4

L2 Or groups get special name (from L1 script), and show bit assignment in 128-bit L2 mask

L2bit: 128 0 l2bit1a
L2bit: **129** 1 l2bit2a
L2bit: 130 **1** l2bit2b
L2bit: 131 2 l2bit3a
L2bit: 132 2 l2bit3b
L2bit: 133 3 l2bit4a
L2bit: 134 3 l2bit4b

EVERY L2 script gets a bit in detailed L2 mask

2nd bit = L1 bit = location in 128-bit mask

L2 128 bit mask = OR of all L2 scripts for corresponding L1 bit

* whether Or'd or Split *

L3bit: 2 128 l3bit1a
L3bit: 3 129 l3bit2a
L3bit: 4 130 l3bit2b
L3bit: 5 2 l3bit3c
L3bit: 6 133 l3bit4a
L3bit: 7 134 l3bit4b
L3bit: 8 3 l3bit4c

Thanks to: Scott Snyder

Python Scripts Reading TDB

These will need maintenance by CD

LET US (Igor and me) KNOW if you are using:

GetBitNames GetRuns GetStuff3

private versions? base on official CD re-releases!

are there any other scripts/programs reading TDB?

More specialized:

Lum DB TDB reader (name?)		(CD/Luiz?)
lmTrigger (currently broken by Lum DB)		(Marco?)
CheckStatCurr	CheckStatUsed	(meisters)
set_statUsed	DeleteTemporaryElements	(meisters)
clientDemo	add_user	(developers)
rcpGen	l3tooltype	(L3?)
Coor resource file makers		(Scott Snyder)

Post Feb 28 Evolution: Some items we will look at

Enhancement of User Interface for large trigger lists using Split/Or

Extension of Trigger Transformation Tool to Removal of terms

Other items dependent on no-server technology will be revisited

Summary

Run2b tag	done
NoLum	testing
Split/Or	
xml generation	testing
user interface	
small trigger lists	now
large trigger lists	in discussion
Trigger Transformation	coding
(with server replacement)	coding

We believe we will have a useable TDB (all required items) for entering commissioning scripts by Feb 28.

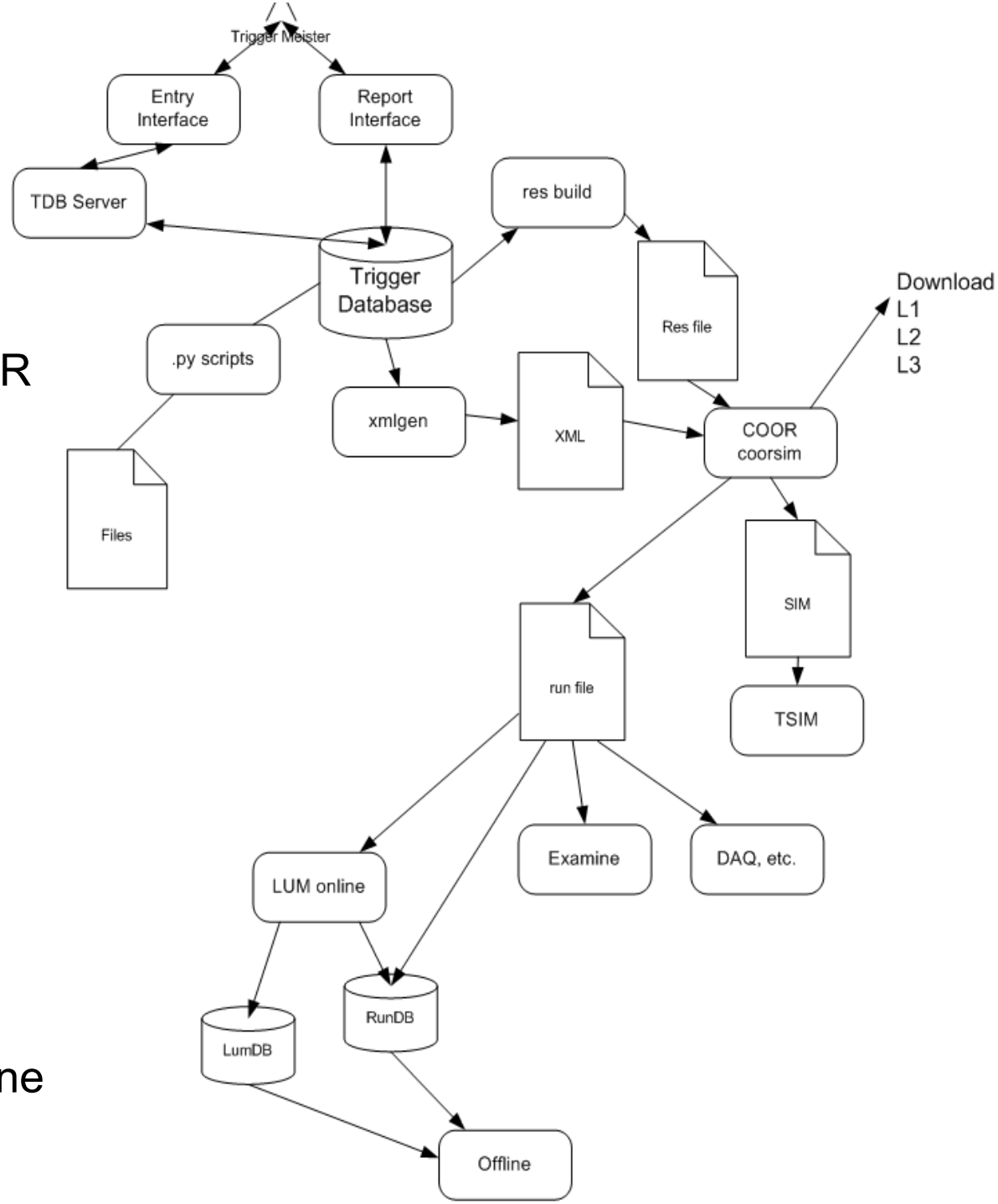
Extra Details for Aficionados (Backup Slides)

Consequences of TDB Implementation of Splitting/Oring

Depends on program's interface to TDB

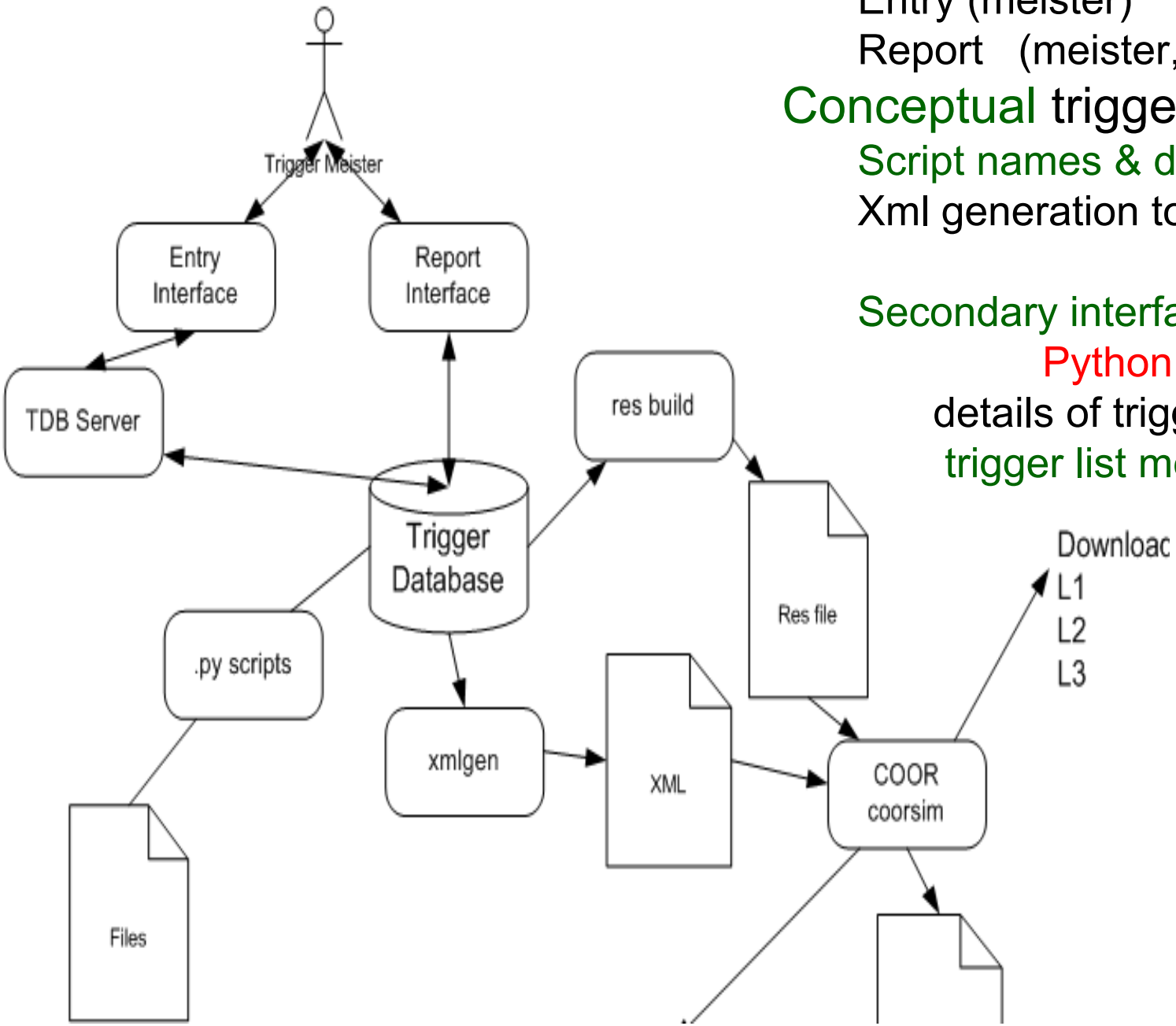
TDB serves:

- Entry
- Report
- Conceptual trigger
- Script names & details
- Xml generation to COOR
- Python scripts



Coor run file serves:

- Bit assignments
  - Trigger Names to bits
  - L1: L2: L3 bits
- Serves this to online
- And via RunsDB to offline



TDB gives:

Entry (meister)

Report (meister, analysts)

Conceptual trigger

Script names & details

Xml generation to COOR

Secondary interface for

Python scripts:

details of trigger

trigger list membership

An example: Conceptual Trigger, XML, and COOR output

l1bit1 -- no splitting or or-ing
- l2bit1a
  - l3bit1a

l1bit2 -- splitting, no or-ing
- l2bit2a
  - l3bit2a
- l2bit2b
  - l3bit2b

l1bit3 -- or-ing, no splitting
- l2bit3a
- l2bit3b
- l3bit3c

l1bit4 -- both or-ing and splitting
- l2bit4a
  - l3bit4a
- l2bit4b
  - l3bit4b
- l3bit4c

Thanks to: Scott Snyder

Example XML to COOR

(written by hand)

2 L2 triggers
An L3 (or more) associated with L1 (or OR of the L2's)

Here 2 L2's also have an L3 script or scripts,

AND a L3 associated with L1

(or OR of all L2's)

```
  <l3trigger name="l3bit2a"/>
 </l2trigger>
 <l2trigger name="l2bit2b">
  <l3trigger name="l3bit2b"/>
 </l2trigger>
</l1trigger>

<!-- or, no split -->
<l1trigger name="l1bit3">
 <l1termlist/>
 <l2trigger name="l2bit3a"/>
 <l2trigger name="l2bit3b"/>
 <l3trigger name="l3bit3c"/>
</l1trigger>

<!-- both split and or -->
<l1trigger name="l1bit4">
 <l1termlist/>
 <l2trigger name="l2bit4a">
  <l3trigger name="l3bit4a"/>
 </l2trigger>

 <l2trigger name="l2bit4b">
  <l3trigger name="l3bit4b"/>
 </l2trigger>

 <l3trigger name="l3bit4c"/>
</l1trigger>
```

Thanks to: Scott Snyder

Coor Run File

Generated from hand-written xml

L2bit: **2 2** L2OR-l1bit3
L2bit: **3 3** L2OR-l1bit4

L2 Or groups get special name (from L1 script), and show bit assignment in 128-bit L2 mask

L2bit: 128 0 l2bit1a
L2bit: **129** 1 l2bit2a
L2bit: 130 **1** l2bit2b
L2bit: 131 2 l2bit3a
L2bit: 132 2 l2bit3b
L2bit: 133 3 l2bit4a
L2bit: 134 3 l2bit4b

EVERY L2 script gets a bit in detailed L2 mask

2nd bit = L1 bit = location in 128-bit mask

L2 128 bit mask = OR of all L2 scripts for corresponding L1 bit

* whether Or'd or Split *

L3bit: 2 128 l3bit1a
L3bit: 3 129 l3bit2a
L3bit: 4 130 l3bit2b
L3bit: 5 2 l3bit3c
L3bit: 6 133 l3bit4a
L3bit: 7 134 l3bit4b
L3bit: 8 3 l3bit4c

Thanks to: Scott Snyder

Some Significant Decisions

Large requirements list: show scope

Splitting and oring within a trigger list/bit

Joint CD/D0 maintenance of xmlgen

Task force effort post Feb 28 agreed by CD

- more is possible w/ server replaced
- including user interface improvement
- and better long term maintenance

Splitting and oring implemented in TDB as is

Some xml-generated test xml (now checking for correctness)


```
<l1trigger name='Afastz_ncu' prescale='0'>
    <l1termlist>
      <l1specterm name='fastz'/>
      <l1specterm name='live_accel_bx'/>
      <l1specterm require='veto' name='skip_next_n_0'/>
      <l1specterm require='veto' name='caltc00'/>
    </l1termlist>
    <l2trigger name='1^min_bias_nim_NCU'>
      <l3trigger name='min_bias_nim_NCU'/>
    </l2trigger>
  </l1trigger>
  <l1trigger name='ALiveBX_ncu' prescale='3400001'>
    <l1termlist>
      <l1specterm require='veto' name='caltc00'/>
      <l1specterm name='live_accel_bx'/>
      <l1specterm require='veto' name='skip_next_n_0'/>
    </l1termlist>
    <l2trigger name='2^zero_bias_NCU'>
      <l3trigger name='zero_bias_NCU'/>
    </l2trigger>
  </l1trigger>
  <l1trigger name='TTK(1,10.)_CEM(2,3)CEM(1,9)_ncu' prescale='0'>
    <l1termlist>
      <l1ctt name='TTK(1,10.)'/>
      <l1emcount count='1' hadveto_refset='null' em_refset='em9'/>
      <l1emcount count='2' hadveto_refset='null' em_refset='em3'/>
      <l1specterm require='veto' name='skip_next_n_0'/>
      <l1specterm name='live_accel_bx'/>
      <l1specterm require='veto' name='caltc00'/>
    </l1termlist>
    <l2trigger name='3^E15_2L20_L25'>
      <l2script>
<l2filter count='1'
name='EM(0,0.,.2,9.,50,EM(0,120.,999.,999.,999.,999.,1.,1.,0,none,5,0,3,3,50))'/>
        <l2filter count='1'
name='HT(0,15.,1,EM(0,0.,1.,1.,2,EM(0,120.,999.,999.,999.,999.,1.,1.,0,none,5,0,
3,3,50)),PASS100,PASS100)'/>
      </l2script>
      <l3trigger name='E15_2L20_L25'/>
      <l3trigger name='E15_2SH10_SH15'/>
    </l2trigger>
    <l2trigger name='5^E16_2L10T0L10L15'>
      <l2script>
        <l2filter count='1'
name='EM(0,0.,.2,9.,50,EM(0,120.,999.,999.,999.,999.,1.,1.,0,none,5,0,3,3,50))'/>
        <l2filter count='1'
name='ETA(0,40,120,EM(0,0.,1.,3.,50,EM(0,120.,999.,999.,999.,999.,1.,1.,1,TRA
CK(0,5.,0,0.,0.,TRACK(0,5.,PTCTT,0,0,0,32,0.)),5,0,3,3,50)))'/>
      </l2script>
      <l3trigger name='E16_2L10T0L10L15'/>
    </l2trigger>
    <l2trigger name='6^ARNOLD_OR1'>
      <l2script>
        <l2filter count='1'
name='EM(0,0.,1.,15.,50,EM(0,6.,0.,0.,0.,0.,1.,6.,0,none,5,0,3,3,50))'/>
      </l2script>
      <l3trigger name='ARNOLD_OR1'/>
    </l2trigger>
    <l2trigger name='7^ARNOLD_OR2'>
      <l2script>
        <l2filter count='1'
name='HT(0,18.,1,EM(0,0.,1.,1.,2,EM(0,120.,999.,999.,999.,999.,1.,1.,0,none,5,0,
3,3,50)),PASS100,PASS100)'/>
      </l2script>
      <l3trigger name='ARNOLD_OR2'/>
    </l2trigger>
    <l3trigger name='ARNOLD_OR3'/>
    <l3trigger name='ARNOLD_OR4'/>
  </l1trigger>
```

Some Operations Issues

DB support, servers, backup etc: CD

TDB software releases (“cut”s)

with D0 Run Coordinator Agreement) CD

DB “operations” (SQL or other specialized tasks) CD

unless new user interfaces provided for meisters

Expert Entry of new trigger elements D0? Who?

Maintaining TDB Status entries should be meisters

CD help in case of problems? EG now

Expert advice & support for triggermeisters (“1:30 am”)

which specific tasks?

who (division between CD and D0; pagers)

Xmlgen: CD + Arnold

Runsum: Operations (+CD?)

StreamXmlGen Adam Lyon/CD?

Python TDB scripts: CD unless D0 request

More Operations Issues: Splitting and Oring

Appropriate Monitoring of new L2 complexity

Effects on online, offline software

- Offline Oring changes inevitable Marco?
- Sensitivity to TDB implementation:
  - Appears to be mainly python scripts (CD)

Testing of new functionality during shutdown

- Nolum should be tested before shutdown

Adequate support of V15 trigger list development (User Interface)