Trigger Menus and Trigger Configuration
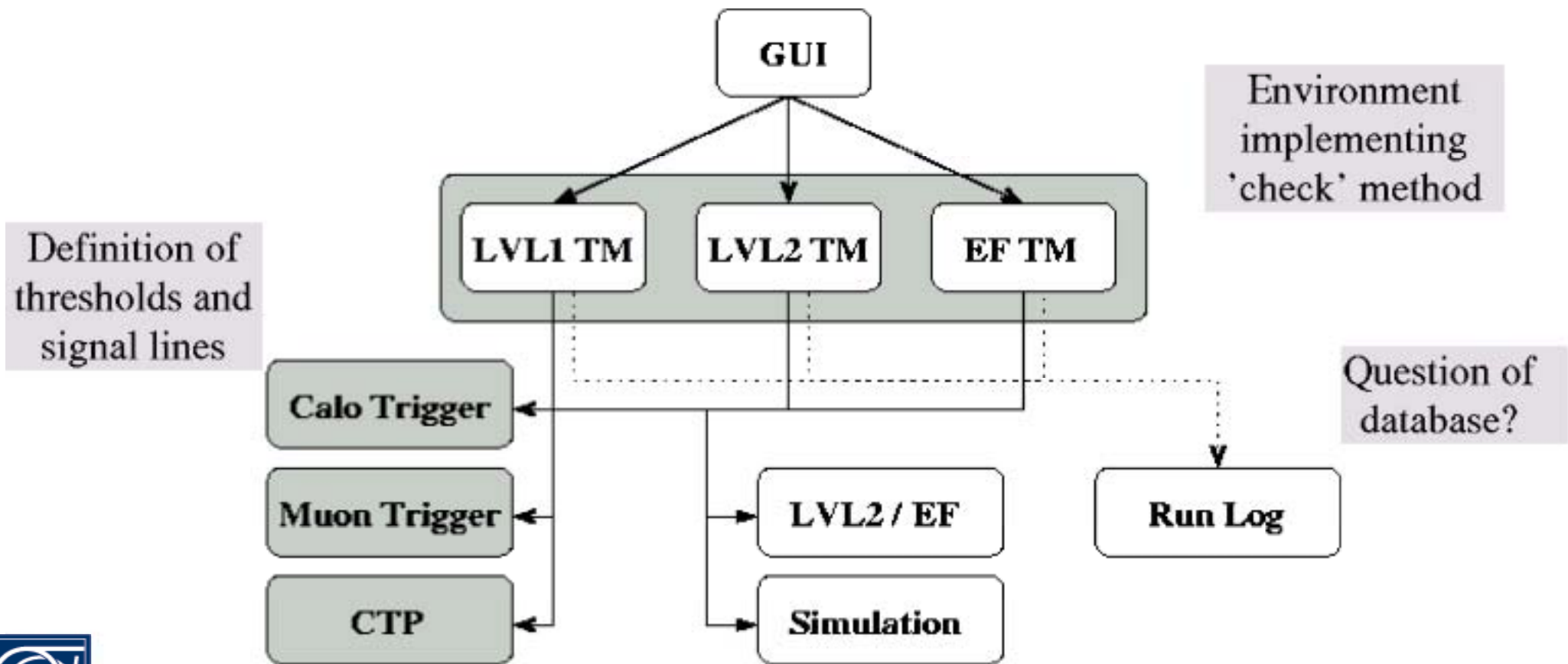
Thomas Schoerner–Sadenius
CERN–EP/ATR

Atlas Calorimeter Trigger Meeting
Heidelberg, 13.–17. 3. 2002

The Problem

➔ We need consistency between HLT/LVL1 trigger menus and hardware configurations.

➔ Trigger configuration should be easy and transparent.

Thomas Schoerner–Sadenius: Trigger Configuration

One Approach ...

➜ LVL1 (software) configuration much easier than HLT

➜ Possibly very difficult to keep consistency when starting with LVL1

==> start with top–level description of HLT configiguration (physics signatures seen by shift crew).

e.g. '$\gamma 40i + \gamma 25i$' or '$e15i + mu20$'

➜ Using a–priori knowledge of HLT abilities ('algorithms') break down phsyics signatures to the elements needed by HLT:

==> sequence tables, menu tables for all steps.

Algorithms: $TE\_out = <TE\_in \mid algorithm>$

➜ But first shortly introduce HLT working principle (took me some time and many discussions to understand ;–)) ...

HLT Principle

Thomas Schoerner–Sadenius: Trigger Configuration
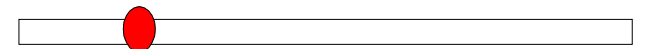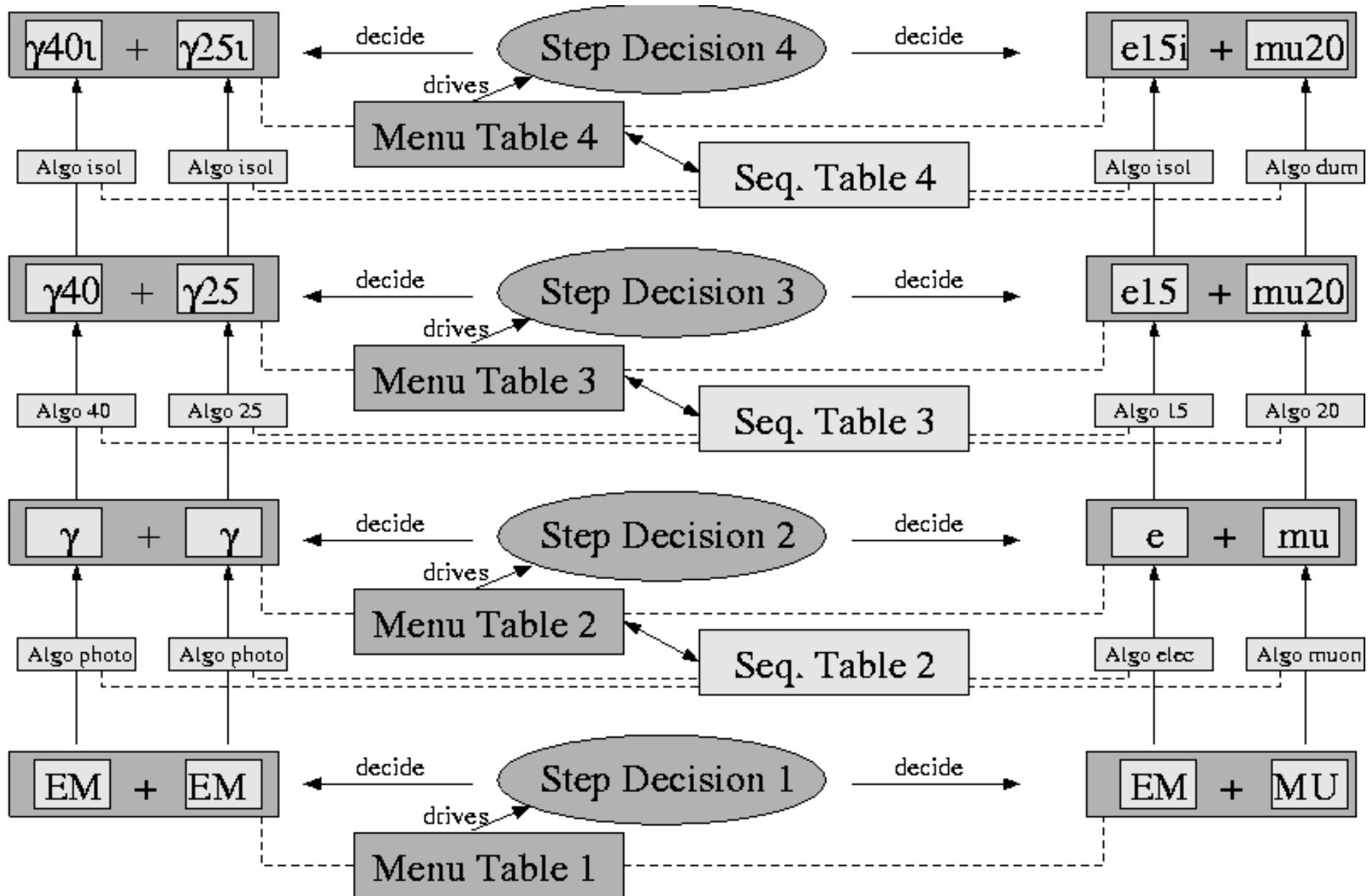
So what we need ...

➔ ... is, for each step, a menu table and a sequence table.

Defines the step signatures which the step decision is looking for.

Sequence Table:

MenuTable:

Defines how to arrive at trigger elements requested by next step signature.

Step 4:

<g40|isol>=g40i
<g25|isol>=g25i
<e15|isol>=e15i
<mu20|dum>=mu20

g40i + g25i
e15i + mu20

Step 3:

<g|40>=g40
<g|25>=g25
<e|15>=e15
<mu|20>=mu20

g40 + g25
e15 + mu20

Step 2:

<EM|photo>=g
<EM|elec>=e
<MU|muon>=mu

g + g
e + mu

Thomas Schoerner–Sadenius: Trigger Configuration

A generic algorithm ...

➔ ... for deriving sequence/menu tables when physics signatures and list of algorithms in/outputs available
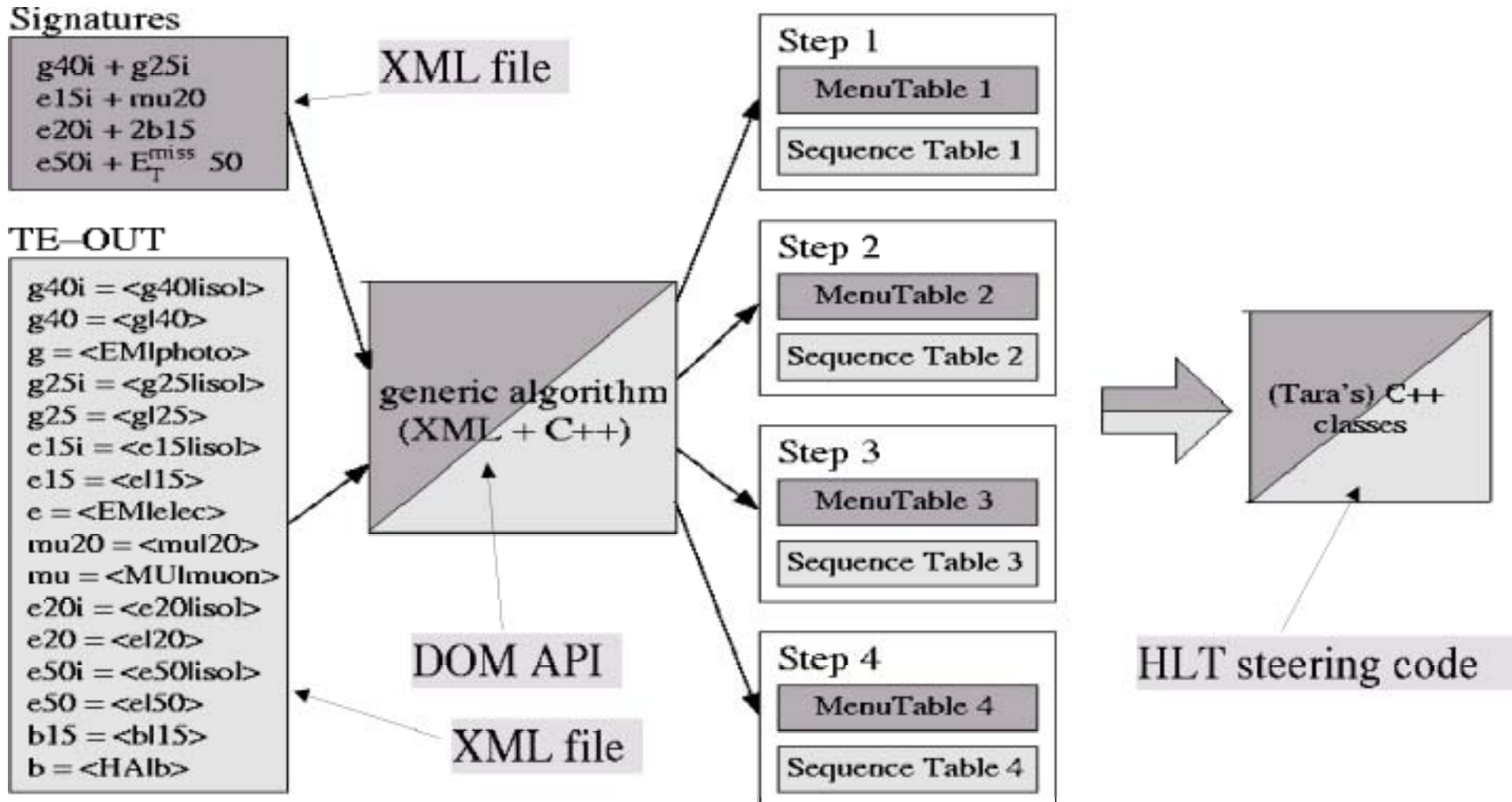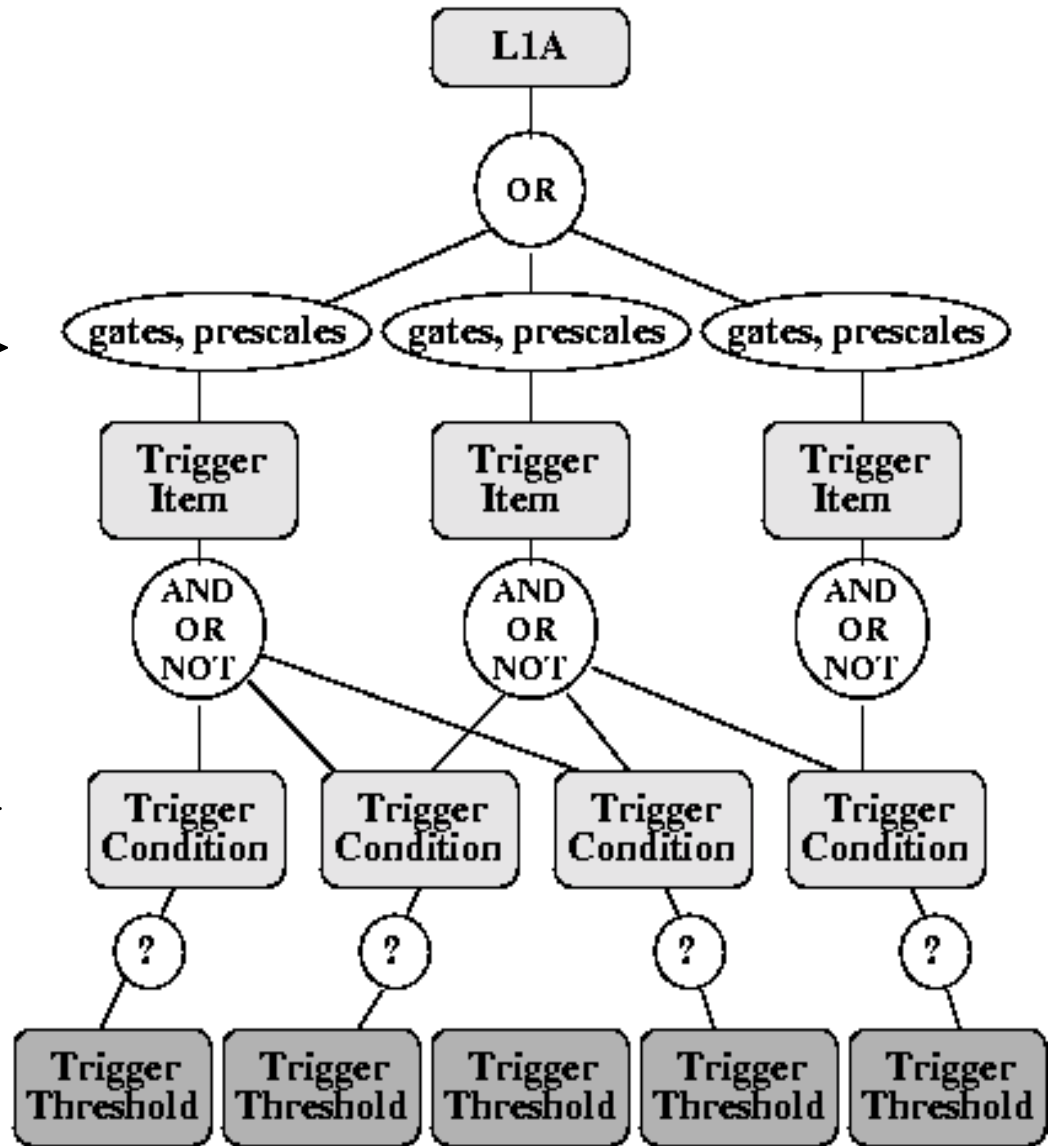
LVL1 Trigger Menus

➔ ... should easily be derived consistently from HLT sequence/menu tables.

➔ Currently stand–alone LVL1 trigger menus for use in simulation of LVL1 central trigger processor (CTP).

  ➔ Also based on XML.

  ➔ Involves also XML hardware description for configuration of LUTs / CMBs on CTP.

  ➔ Issue of consistent calo/muon trigger configuration.

  ➔ Starting integration with Ed's calo simulation within Athena.
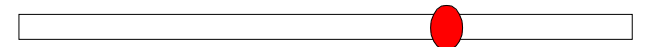
XML LVL1 Trigger Menus

\<TriggerMenu TM_ID="tdr"\>

\<Item TI_ID="MU1" mask="on"
   priority="low" prescale="10"\>

    \<Condition thresh="MU3" mult="1" /\>

\</TriggerItem\>

\</TriggerMenu\>

\<ThresholdList\>

\<Threshold name="MU3" type="MU"\>

    \<ThresholdValue thresholdval="6" ...\>

\</Threshold\>

\</ThresholdList\>

Problems, Questions, Challenges

➔ Top–down approach really fully feasible?

    ➔ Threshold definitions?

    ➔ What does HLT have to know about L1?

➔ Prescales:

    ➔ Since we want fast reject in HLT, prescales should be applied asap, e.g. at the lowest step possible. But what if one step signature is used by two or more physics signatures with different prescales?

➔ Code that generates menu/sequence tables is running as standalone. Now integration with trigger steering/ configuration software needed (in fact, it has just started and is not completely straight forward).

➔ ... (I am sure there are more).

Interface

L1HLTSequenceTable* getSequenceTables()

```
class L1HLTSequenceTable {

    public:      L1HLTSequenceTable();
                 ~L1HLTSequenceTable();
                 vector<L1HLTSequence*>& sequenceVector() const;

    private:     vector<L1HLTSequence*> m_L1HLTSTSequenceVector;

};
```

L1HLTMenuTable* getMenuTables()

```
class L1HLTMenuTable {

    public:      L1HLTMenuTable();
                 ~L1HLTMenuTable();
                 vector<L1HLTSignature*>& signatureVector() const;

    private:     vector<L1HLTSignature*> m_L1HLTMTSignatureVector;

};
```