The Configuration System of the ATLAS Trigger

Andre dos Anjos*, Paul Bell†, David Berge‡, Johannes Haller§¶, Simon Head†, Shumin Li¶,
Andreas Hoecker‡, Takanori Kohno‡, Tania McMahon||, Miroslav Nozicka¶, Hans von der Schmitt**,
Ralf Spiwoks‡, Joerg Stelzer‡, Thorsten Wengler†, Werner Wiedenmann*
*University of Wisconsin, Madison, USA, †University of Manchester, UK, ‡CERN, Geneva, Switzerland,
§Hamburg University, Germany, ¶DESY, Hamburg, Germany, ||Royal Holloway and Westfield College, London, UK,
**Max-Plack-Institute, Munich, Germany

Abstract—**The ATLAS detector at CERN's LHC will be exposed to proton-proton collisions at a rate of 40 MHz. To reduce the data rate, only potentially interesting events are selected by a three-level trigger system. The first level is implemented in custom-made electronics, reducing the data output rate to less than 100 kHz. The second and third level are software triggers with a final output rate of 100 to 200 Hz. A system has been designed and implemented that holds and records the configuration information of all three trigger levels at a centrally maintained location. This system provides consistent configuration information to the online trigger for the purpose of data taking as well as to the offline trigger simulation. The use of relational database technology provides a means of reliable recording of the trigger configuration history over the lifetime of the experiment. Tools for flexible browsing of trigger configurations, and for their distribution across the ATLAS reconstruction sites have been developed. The usability of this design has been demonstrated in dedicated configuration tests of the ATLAS level-1 Central Trigger and of a 600-node software trigger computing farm. Further tests on a computing cluster which is part of the final high level trigger system were also successful.**

I. Introduction

Scheduled to deliver first physics data in early 2008 the Large Hadron Collider (LHC) at CERN will collide protons at a center-of-mass energy of 14 TeV with luminosities of up to $10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$. The ATLAS detector is designed and built to record the proton collisions. The research program comprises the search for the Higgs boson and for supersymmetry, precision measurements of the W-boson and top-quark properties, and B-meson physics. To achieve these goals a detailed understanding of the trigger and its performance is essential.

The ATLAS trigger is a complex and highly configurable system. For data taking the complete configuration of the trigger hardware and software must be provided fast and consistently. To reproduce, study, and understand the trigger behavior for all taken data, knowledge of the trigger configuration at any point in time must be available. The access to this information should be simple and user friendly. Its usage must be consistent for data-taking and for offline trigger simulation and studies. The ATLAS trigger configuration system has been designed to satisfy these requirements.

II. The ATLAS Trigger

At the LHC proton bunches will cross at a rate of 40 MHz. A highly selective and efficient trigger system identifies and accepts interesting physics events, while reducing the overall data rate to levels that fit within the constrains of the data-flow and storage systems. The ATLAS trigger is a three-level system with a final output rate of about 200 Hz.

The first level (LVL1) [1] is implemented in custom-made electronics, reducing the event rate to less than 100 kHz. Within a latency of $2.5\,\mu$s, LVL1 makes its decision based on comparatively coarse objects which it receives from the calorimeter and the muon trigger electronics. This information is combined in the central trigger processor (CTP); the LVL1 accept decision is purely based on the multiplicities of these muon and calorimeter trigger objects. In addition the LVL1 can issue triggers on selected or random bunch-crossings. The High Level Trigger (HLT) [2], which consists of the Level 2 (LVL2) and the Event Filter (EF) trigger, is implemented in software and is executed on a farm of about 2000 rack-mounted computing nodes. HLT algorithms have access to data from all ATLAS sub-detectors. In LVL2, algorithms operate on 'Regions of Interest (RoI)' – regions of the detector confined in terms of pseudorapidity (η) and azimuth (ϕ) – which correspond to the LVL1 trigger objects. A LVL2-accept signal initiates the event building process where information from all RoIs gets collected and passed on to the EF processes; EF algorithms use the entire event data to derive their decision. Events accepted by the EF are written to mass storage. The LVL2 time budget is about $10\,\mu$s, the EF budget a few seconds.

A. The Level 1 Trigger Event Selection

The LVL1 trigger makes the initial event selection based on muon trigger detectors and reduced-granularity calorimeter information. The calorimeter selections are based on information from all the ATLAS calorimeters (electromagnetic and hadronic; barrel, endcap, and forward). The LVL1 Calorimeter Trigger aims to identify high transverse-energy (E_T) electrons and photons, jets, and hadronically decaying tau leptons, as well as events with large missing and total transverse energy. A trigger on the scalar sum of jet transverse energies is also available. For the electron/photon and tau/hadron triggers, isolation can be required. Multiplicity information is available for a number of programmable E_T thresholds (between 4 and 16 thresholds) per object type. The LVL1 Muon Trigger is based on signals seen in the muon trigger chambers: resistive-plate chambers (RPCs) in the barrel, and thin-gap chambers (TGCs) in the endcaps. The trigger searches for patterns of

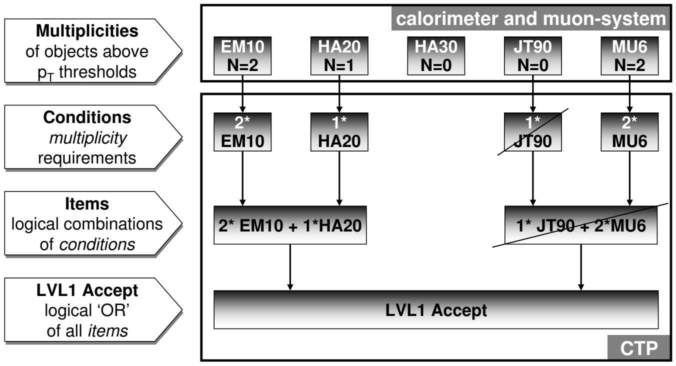

Fig. 1. Level 1 trigger selection strategy. Trigger objects are built by the muon-system and calorimeter trigger hardware. Their numbers are counted and the multiplicities are checked against the requirements of the trigger item. A satisfied requirement that also survives the prescaling issues a LVL1 accept. The complete collection of trigger items form the LVL1 part of the trigger menu.

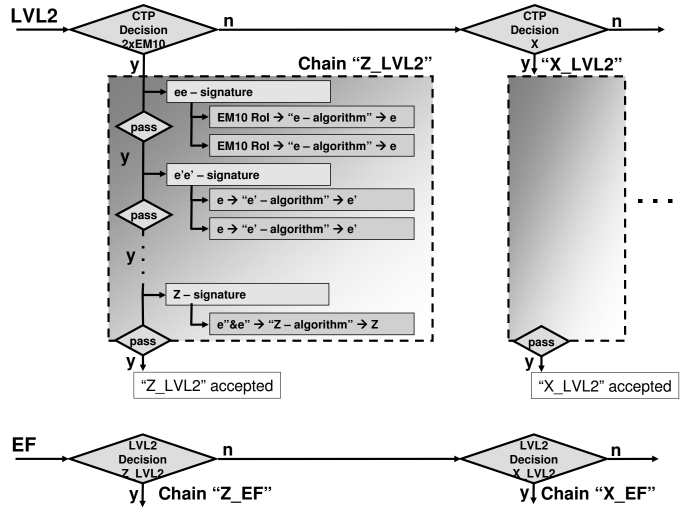

Fig. 2. The high level trigger event selection strategy. In both levels (LVL2 and EF) it follows the concept of trigger chains that define independent event selections. A chain is an ordered list of signatures, *e.g.* “two electrons” or “one Z candidate”, that are evaluated in sequence. The complete collection of chains form the HLT part of the trigger menu.

hits consistent with high transverse-momentum (p_T) muons originating from the interaction region. The logic provides six independently-programmable p_T thresholds. The information for each bunch crossing used in the LVL1 trigger decision is the multiplicity of muons for each of the six p_T thresholds. The overall LVL1 accept decision is made by the CTP by combining the information for different object types into *LVL1 trigger items*, *e.g.* `2*EM10+1*HA20`, (Fig. 1). Trigger menus can be programmed with up to 256 distinct items, each item being a combination of requirements on the input data. The possibility of lowering the rate of a certain trigger item by applying a scaling factor (LVL1 prescale) is built into the system. When configuring the LVL1 trigger constraints given by the trigger electronics have to be respected. These constraints concern not only the number of available thresholds, but also the available combinations of their values and the consistency of configuration parameters and physical connectivity in the system.

Upon the event being accepted by the LVL1 trigger, both the muon and the calorimeter trigger processors send geometric location of trigger objects (RoIs) to the LVL2 trigger, where it is used to guide the processing of data preceding the final trigger decision. The trigger acceptance pattern (256 bit) is stored in the event.

B. The High Level Trigger Event Selection

The HLT event selection in done in a *step-wise* and *seeded* manner. At each step the selection is refined using information from increasingly more subdetectors. The HLT trigger selection is described by the HLT trigger configuration, which is stable within a data run or a run-block.

The full HLT trigger configuration is composed of the *trigger menu* and the *trigger setup*. The menu describes the trigger selection logic, the setup contains the individual configuration parameters of all trigger algorithms and additional software components. At each level the trigger menu is composed of *trigger chains*, each chain defining a particular physics trigger (Fig. 2). A chain is an ordered list of *trigger signatures* – each representing a step – that are evaluated sequentially. A trigger signature at a certain step is the logical combination of *trigger elements*, *e.g.* two electrons, or a Z-candidate. A chain in LVL2 is executed if the prerequisite LVL1 item has been satisfied. Execution of an EF chain depends upon the success of the prior LVL2 chain. Following the *early reject* trigger principle, chain execution proceeds up to the step where a signature is not satisfied, at that point the chain is deactivated for the current event.

The trigger accept rate can be adjusted by changing the HLT scale factors (HLT prescale set), possibly for each trigger chain at both levels. For simplicity there are no prescale factors at the intermediate steps of a chain. An active chain that passes the prescaling will cause the event to be kept for storage and later analysis. For accepted events the list of satisfied trigger chains at each level is stored in the trigger results (equivalent to the trigger bit pattern in LVL1).

The trigger elements that are the building blocks of the signatures are connected to sets of trigger algorithms. Such a set usually consists of one or more *feature extraction algorithms* that create trigger objects such as tracks or clusters and *hypothesis algorithms* that validate the trigger elements based on properties of the trigger objects.

Algorithms have access to trigger elements produced in earlier steps and can request additional data from the Read-Out System (ROS). Hence the selection in the HLT is *seeded*. The initial seeds of the LVL2 selection are the RoIs identified by LVL1. These are refined from step to step by the HLT algorithms. The division of trigger chains into several steps per level allows to first execute those trigger algorithms that have large rejection power and need little processing time and data movement (note that for LVL2 the full granularity data in the RoIs are transfered from the ROS to the HLT processing node). So, for instance, does the electron trigger execute the cluster algorithms and cluster based selections before the track reconstruction is performed.

In LVL2 algorithms usually request full granularity data around the initial RoI from a certain subdetector only, leading to a significant reduction of data transfer. The EF is executed after the event building and has access to data of the full event.

The algorithmic processing of the HLT is controlled by the HLT steering software. It is responsible for all framework functionality of the HLT selection: the activation and termination of trigger chains, the step-wise testing of signatures, the combinatorics of trigger objects as input to the algorithms, and the algorithm execution. Thereby a caching mechanism prevents multiple execution of the same algorithms on the same input data. The HLT steering is described in detail in [3].

C. Trigger Configuration Data

Based on the selection strategy of the three trigger levels, various configuration data are needed. For LVL1 the trigger configuration data consist of

- the trigger menu definition; the list of LVL1 items in the CTP, i.e. the logical combination of requested calorimeter and muon multiplicities,
- the definition of trigger thresholds for which the calorimeter (E_{T}, isolation) and muon (p_{T}) trigger hardware delivers multiplicities to the CTP,
- technical parameters like prescale factors for the LVL1 items, dead-time parameters, and
- internal parameters like random trigger rates.

In addition, the human-readable definition of the trigger menu has a representation that matches the layout of the CTP hardware. This representation is built by the *trigger menu compiler* and is also part of the configuration data.

The HLT configuration consists of

- the trigger menu definition; the list of LVL2 and EF trigger chains and their decomposition into signatures with the corresponding multiplicity requirements for the created trigger elements,
- the link between the trigger elements and the algorithms that create them, and
- the specific configuration parameters of trigger algorithms and other software components,
- LVL2 and EF prescale factors that are applied separately to each chain, and
- a string identifying the software release used with this configuration.

III. The Trigger Configuration System

Information about current and past trigger configurations is requested by several ATLAS clients: online data taking, trigger simulation, and data analysis. The amount of information differs from client to client. This leads to the following requirements on the ATLAS trigger configuration system.

1) Complete and consistent configuration of the ATLAS trigger.
    - Configuration of the online trigger hardware and software for data taking.
    - Configuration of the trigger simulation software for Monte Carlo production.
2) Distribution of configuration information for the physicist to perform trigger aware analysis and trigger studies.
3) Fast and flexible adjustments of the trigger menu and setup in response to changing beam and detector conditions during data-taking.
4) Configuration storage, providing a history of all current and past trigger configurations.

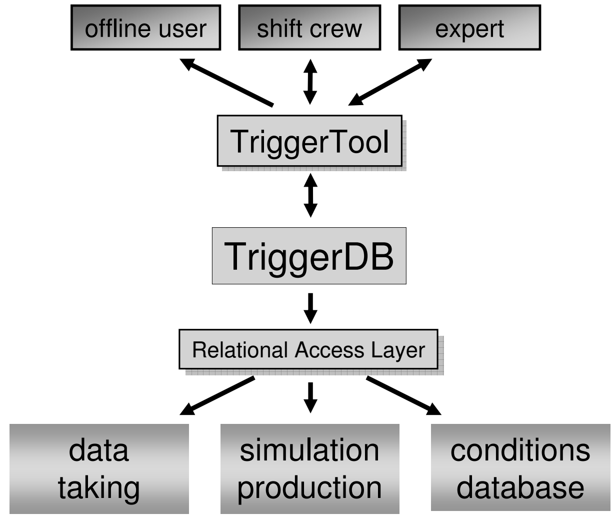

Fig. 3. Schematic overview of the ATLAS trigger configuration system. The core is a relational database (TriggerDB), which the user can interact with through a Java based GUI (TriggerTool). Clients can also directly interact with the database to extract information necessary to configure the trigger for data taking and for simulation, or information storage in the conditions database.

The ATLAS trigger configuration system has been designed to meet these requirements. It consists of four components (Fig. 3):

- the configuration database (TriggerDB) in which trigger configurations are stored,
- the graphical user interface (TriggerTool) to create new and manipulate existing trigger configurations,
- the database client software that retrieves configuration information necessary for data-taking and trigger simulation, and
- distribution software to provide the configuration information for prompt reconstruction and data analysis.

A. The Trigger Database

The TriggerDB is the central part of the configuration system. It is used to store and preserve all data that are needed to configure the three levels of the ATLAS trigger. The TriggerDB is implemented as a relational database. The supported technologies are Oracle, used during data taking at the site of the experiment, MySQL for simulation production at computing sites without Oracle support, and SQLite for local storage of configurations, *e.g.* the personal laptop. The different elements of the configuration are stored in different tables of the database. The relational structure links the various elements in a hierarchical order. So it defines *e.g.* which parameters belong to an HLT algorithm, or which LVL1 trigger items are part of the LVL1 trigger menu. The size of the database is kept small by reusing parts of the data in different configurations and hence avoiding unnecessary data duplication. For the purpose of human readability all configuration elements are given a string name and a version number.

At the top level the LVL1 and HLT trigger menus are linked to the master table. The identifier of this table is the unique integer key by which the complete configuration (with the exception of the trigger prescale values) can be unambiguously accessed. The sets of prescale values for the LVL1 and the HLT trigger decisions are expected to change more frequently during data taking and therefore have their own identification key. Thus three keys are needed to define the full trigger behavior.

It is foreseen to store all configurations that have been used for data-taking, trigger simulation, and test runs in the TriggerDB. The three configuration keys will, for each run, be transfered to the ATLAS conditions database (COOL). These keys provide the unique reference to the trigger configuration for each run and can therefore be used to retrieve the configuration at a later stage. A locking mechanism for used configurations is in place, entries in the TriggerDB that are part of a used configuration are marked as such and protected. Thus, the complete trigger configuration history over the lifetime of the ATLAS experiment is recorded.

The consistency of the configuration data is an essential requirement that the configuration system must fulfill. Inconsistent trigger configurations can lead to data loss or data unusable for physics analysis. Wherever possible, the relational schema has been designed to enforce consistency. Moreover, the database population and manipulation tools scrutinize the consistency of the data they handle.

B. The Graphical User Interface

The TriggerTool is a graphical user interface that has been developed to access the TriggerDB. The TriggerTool allows browsing and manipulating of configuration data. Its primary role is to support convenient, fast, and flexible manipulation of the trigger configuration in response to changing beam or detector conditions during data taking. It also safeguards the archiving of all configurations that have been used for any period of data taking, such that they can be retrieved during subsequent offline analysis.

The TriggerTool is a standalone project written in JAVA. It supports access from remote locations to the TriggerDB and its copies for all three database technologies. On the main panel (Fig. 4) a search bar allows the user to retrieve individual configuration elements or whole configurations matching certain criteria. A more sophisticated search panel is also available. Upon selection of a configuration, a tree viewer in the lower left of the main panel shows the hierarchical structure of a trigger menu: the LVL1 trigger items, the HLT chains, signatures, algorithms and parameters. Reflecting the nature of the database schema, the tree viewer can also be used to identify to which configurations a certain configuration element, *e.g.* a chain or algorithm, belongs.

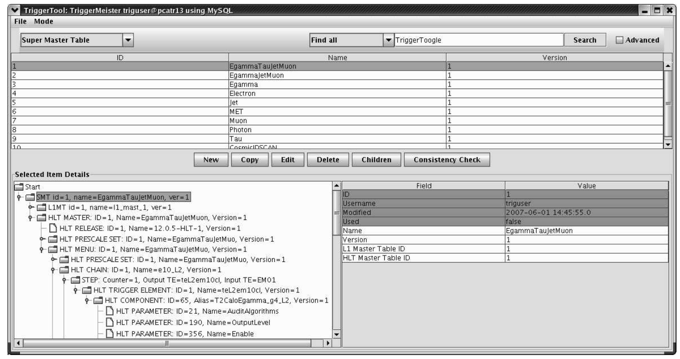

Fig. 4. Screen shot of the TriggerTool in expert mode. From the list of available configurations (top) one was selected and displayed in detail in a tree structure on the bottom left. The tree is expanded to the view the HLT chain (`e10_L2`) and the algorithm (`T2CaloEgamma_g4_L2`) that runs as the first step in that chain. The parameters of this algorithm are also shown. On the bottom right detailed information about the entry selected in the tree is shown. Double-clicking an item in the tree opens a edit window for that entry.

The scope of permitted editing operations depends on the current mode of operation of the TriggerTool. Different rights are implemented for trigger experts, shift operators and offline users. In expert mode the user can upload complete new configurations encoded in extended markup language (XML) format (section III-C). The expert can also create or delete any record and add or remove existing records from higher levels in the tree, thereby allowing, for example, existing LVL1 items or HLT chains to be added to newly created menus. In this way, existing configurations can be changed or entire new ones built up from scratch. Such operations are restricted by various validity checks performed by the TriggerTool. In addition to trivial checks of the form and type of data being entered into individual fields, the TriggerTool also ensures the consistency of the trigger menu within and across the three trigger levels. As part of the locking mechanism for used configurations the TriggerTool also forbids the editing of any record marked as "used". Due to the nature of the database schema, whereby a certain record can appear in multiple configurations, various warnings are implemented to avoid the user inadvertently changing configurations belonging to many configurations. Upon changes of the LVL1 trigger menu, its trigger hardware representation needs to be rebuild. For this purpose the trigger menu compiler is also integrated into TriggerTool.

For the shift mode the emphasis has been put on the easy operation of the trigger during regular data-taking. Most often the shifter will be required to change between different configuration aliases (e.g. defined for physics data, cosmic data, or calibration runs) which each corresponds to a different available master keys. Different alias sets of prescale factors (*e.g.* high, medium, and low luminosity) can also be selected. It will be the job of an expert to assign these shifter aliases to the currently approved combinations of master keys and sets of prescale factors. Occasionally, depending on machine and detector conditions, the shifter might need to add or remove a certain trigger, or raise or lower the value of a trigger selection parameter, so some basic editing is allowed.

The user mode allows no manipulation of the database, only the viewing and comparison of certain configurations or of the history of a particular entry in the TriggerDB (e.g. a particular trigger selection cut).

C. Online Trigger Configuration

Before trigger configurations are used for data taking they are thoroughly tested in an online-like environment. As a final step of the test procedure a snapshot of a running trigger process is taken. The snapshot contains the LVL1 and HLT trigger menu, as well as the values of all configurable parameters of the running trigger. The format of the snapshot is xml which can be uploaded into the database. The upload of such a configuration into the TriggerDB, which will happen prior to data taking using the TriggerTool, returns the three identifier keys (configuration, LVL1 prescale, and HLT prescale sets) that will serve as unique identifiers for later access to this configuration.

The shifter will chose these keys indirectly by selecting one of the available configuration aliases (section III-B). The keys are then published to the online trigger clients – the LVL1 muon-system and calorimeter trigger hardware, the CTP, the HLT configuration handler, and the COOL configuration writer – by the ATLAS run control. Using these keys, the clients access the TriggerDB through a relational access layer (CORAL), which is a C++ front end to the different database technologies. It gives direct access to the relevant information in the TriggerDB.

Having a large number of computing nodes simultaneously connecting to the TriggerDB puts strain on the database server. At ATLAS a database proxy (DBProxy) system has been developed that caches database queries and their results and thus takes the burden off the server.

D. Distribution of Trigger Configuration Information for Data Analysis

Data analysis depends on the knowledge of the configured and accepted triggers in each event, including their prescale rates. While trigger objects and trigger-accept bit-patterns are stored with each event, the configured triggers are run meta-data. This includes the name, version, bit-position in the trigger-accept pattern, and the prescale value for each LVL1 trigger item and HLT trigger chain. Information about the trigger signature at each step of the trigger chain is also part of this data. For the run meta-data a different distribution mechanism has been implemented (Fig. 5). When data is taken, trigger configuration information is written to COOL, which is constantly replicated to the prompt reconstruction sites. During prompt reconstruction the configuration information

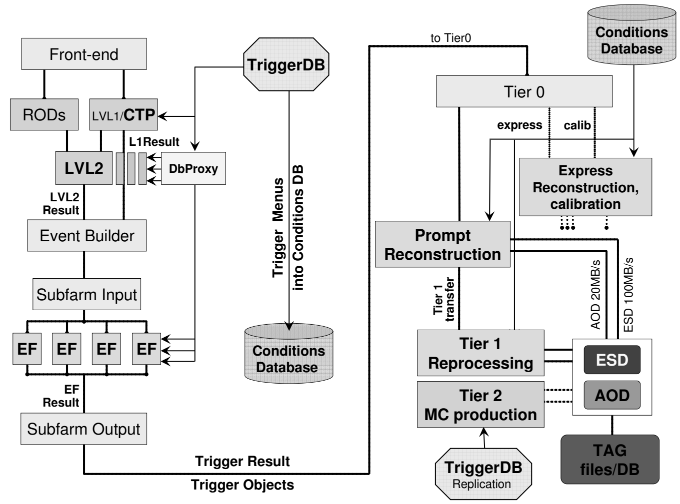

Fig. 5. Flow of configuration data. Primary source of configuration data is the TriggerDB. From there the LVL1 hardware and HLT software is configured for data taking. The trigger menu configuration but no further details such as algorithm parameters or LVL1 thresholds are stored in the conditions database. This information is used during prompt reconstruction to obtain the association between trigger names and trigger bits and store it in the AOD files.

is written as run-wise meta-data into the analysis-object-data (AOD) files. Together with the event-wise trigger decision and trigger objects the physics analyst possesses all information that is necessary to select events based on the name and version of a trigger definition. In addition to the trigger acceptance bit-pattern the index of the last successfully executed step for each trigger chain is stored in the trigger decision, giving the analyst means to study the trigger performance in great detail using AOD information.

In a similar manner trigger-accept patterns and the necessary configuration information to interpret them are stored in reduced *tag files* and an Oracle based *tag database*.

Subsets of the configurations in the central TriggerDB are also replicated to Tier-2 sites for Monte Carlo simulations production.

E. First Tests

The configuration of the LVL1 hardware from the TriggerDB has successfully been tested in the ATLAS cavern in several *technical runs* since October 2006, it is now the default LVL1 configuration method.

The first extensive test of the HLT configuration was successfully performed on a computing farm of about 600 nodes in December 2006. The capability of the system to simultaneously configure the two levels of the HLT with complex trigger menus was demonstrated. In later tests the TriggerTool was employed to quickly change trigger menus or parameters of active trigger configurations. These tests also gave valuable information about the needs of the shifter during data-taking in terms of the configuration of the trigger using the TriggerTool.

During these tests the DBProxy system has successfully been used together with the trigger configuration software.

F. Summary

For the trigger of the ATLAS experiment a system has been designed that provides consistent configuration for all three trigger levels for data taking as well as for the Monte Carlo simulation of the trigger. The system preserves trigger configurations for the lifetime of the experiment. A tool is provided that allows for easy browsing and manipulation of the configuration data. The flow of configuration data to the user for the purpose of physics analysis is defined, the necessary software is implemented.

To a large extent the system has been implemented and tested at the experimental site. Configuration of the LVL1 CTP is included in the run control environment. The HLT configuration system has been deployed on parts of the final HLT processing farm.

REFERENCES

[1] The Atlas Collaboration, "Atlas First-Level Trigger Technical design report", CERN/LHCC/98-14, 1998.

[2] The Atlas Collaboration, "ATLAS High-Level Trigger, Data Acquisition and Controls: Technical Design Report", CERN/LHCC/2003-022, 2003.

[3] N.Berger et al., "The High-Level-Trigger Steering of the ATLAS Experiment", these proceedings.