Workflow Management in the Assembly of CMS ECAL

N. Baker [a], A. Bazan [b], F.Estrella [a], Z. Kovacs [a], T. Le Flour [b], J-M. Le Goff [c], E. Leonardi [d], S. Lieunard [b], R. McClatchey [a], J-P Vialle [b]

[a] *Computing Dept., Univ. of the West of England, Bristol, BS16 1QY, UK,*

[b] *LAPP, IN2P3, Annecy-le-Vieux, France,*

[c] *ECP Division, CERN, Geneva, 1211 Switzerland,*

[d] *University of Roma III, Rome, Italy*

As with all experiments in the LHC era, the Compact Muon Solenoid (CMS) detectors will be constituted of a very large number of constituent parts. Typically, each major detector may be constructed out of over a million precision parts and will be produced and assembled during the next decade by specialised centres distributed world-wide. Each constituent part of each detector must be accurately measured and tested locally prior to its ultimate assembly and integration in the experimental area at CERN. Much of the information collected during this phase will be needed not only to construct the detector, but for its calibration, to facilitate accurate simulation of its performance and to assist in its lifetime maintenance. The CRISTAL [1] system is a prototype being developed to monitor and control the production and assembly process of the CMS Electromagnetic Calorimeter (ECAL). The software will be generic in design and hence reusable for other CMS detector groups. This paper discusses the distributed computing problems and design issues posed by this project. The overall software design architecture is described together with the main technology aspects of linking distributed object oriented databases via CORBA with WWW/Java-based query processing. The paper then concentrates on the design of the workflow management system of CRISTAL.

Key words: distributed systems, workflow management, cooperative task management, production and assembly system.

1 Introduction

The initial phase of the CRISTAL (Concurrent Repository and Information System for the Tracking of Assembly Lifecycles) project is concerned with

the production and tracking of the 110,000 lead tungstate ($PbWO_4$) monocrystals and their fast electronics to be installed in the CMS ECAL detector. Due to the number of crystals involved in ECAL construction and the very high standard to which each must be grown there will be a number of Production Centres located in Russia, China, and the Czech Republic. Assembly of the crystals with their Avalanche Photo-Diodes (APDs) and associated electronics and mountings will take place in so called Regional Centres located in Italy, UK, Russia and at CERN which will also act as the coordination centre. The total time needed for the production of the crystals will be of the order of 5 to 6 years and will commence in 1998. Each of the crystals will have their physical characteristics individually measured and recorded to facilitate calibration and to ensure consistency of the production process. Since the overall costs and timescales of crystal production must be strictly controlled, the efficiency of the production process is paramount. It therefore follows that quality control must be rigidly enforced at each step in the fabrication process. The CRISTAL system must support the testing of detector parts, the archival of accumulated information, controlled access to data, on-line control and monitoring on all Production and Regional Centres.
An Engineering Data Management System (EDMS), or Product Data Management system, will be used to manage, store and control all the information relevant for the conception, construction and exploitation of the LHC accelerator and associated experiments during their complete life cycle estimated to be more than 20 years. All the engineering drawings, blueprints, construction procedures, part definitions and part nominal values will be stored in the EDMS. EDMS is document management-oriented whereas CRISTAL manages aspects related to the control and production of the crystal detector. CRISTAL will need to be able to access EDMS for part definitions and tolerances. Ultimately CRISTAL may be required to provide EDMS with some physics parameters for future upgrades of the experiment. Detector simulation software provides both CRISTAL and EDMS with the ideal parameters for all the detecting devices. During production and assembly CRISTAL will be required to store all the deviations from the ideal specifications. These deviations will be accessed by calibration software for event reconstruction. Some information may also be relevant to the Detector Control System during experiment operation. The relationship of CRISTAL to other software in CMS is investigated in [1].
In summary the CRISTAL project aims to implement a prototype distributed engineering information management and control system which will control the production process of the crystal detector and provide secure access to calibration and production data. The specific objectives are to:-

- design and build a distributed information management system to control and monitor crystal production across all centres
- capture and store crystal calibration data during detector production lifecycle

- provide detector construction with quality control and assembly optimisation data
- integrate instruments used to characterise parts and provide controlled, multi-user access to the production management system
- provide access to engineering and calibration data for CMS users and physics programmes

2 Overall Architecture & Design

Management information regarding the definition, configuration, version, performance and operational state of the distributed CRISTAL production line is stored in a central repository. This object based central repository also stores the definitions of all the parts that make up the detector together with the definitions of the instruments used to produce parts or take measurements of parts. It also stores descriptions of the life-cycle of each part and descriptions of the tasks and activities performed on the part. Each physical part is allocated a unique identifier (bar code) when it is produced. The unique identifier is used as a reference to an object description of the part in the central repository. The part object not only stores the current state and characteristics of the physical part but holds a reference to its current position in the production life cycle and a reference to its possible future production flows.
A part identifier is therefore used by a production operator to recall its life history and to provide navigational assistance as to the next possible sequence of tasks that could be executed. This production scheme (or sequence of tasks) determines the order of tasks that can be applied to a part. Each type of part has a different production scheme. The human operator can trigger the execution of a task. The task execution script will run and, via a console, prompt the operator to perform a number of manual operations, such as cleaning the part, or it may automatically trigger networked instruments to take measurements of the physical characteristics of the part. All the details of the task operations and the associated measurements and instruments involved are eventually stored in the central repository. A part can be defined as a collection of parts, a necessary condition as the detector is gradually assembled. Over time the part and task definitions will evolve as a result of knowledge that emerges during detector testing and construction. A major critical design requirement is that the work flow management system of CRISTAL should follow and guide the environment instead of imposing constraints on the users. The analysis and design team has spent a considerable amount of time capturing user requirements. These requirements have been published in a User Requirements Document which conforms to the PSS-05 Software Engineering Standards [2] defined by the European Space Agency.
The data handling and storage aspects of CRISTAL must be transparent to any of the users of the system irrespective of location. Part, task, production schema and system configuration data will be defined by the Coordinator at

the CERN central site. The Coordinator will determine when these definitions become active and when they are distributed to the Regional and Production Centres. The configuration and management of all the centres is controlled by the CERN central site. Once the production centres are registered and configured parts will be produced and then shipped to further centres for testing and assembly. As the physical parts migrate through the production, testing and assembly life cycle so rapid access will be required to the part objects which define the state of the part and hold references to its history and measured characteristics. Since the projected final amount of data contained in CRISTAL is of the order of 1 Terabyte, a replicated database approach with 10 centres is not technically or economically feasible. The strategy being adopted is to only store objects in a centre repository or database which are directly related to physical parts held locally, but retain a regularly updated centralised database at CERN. As a consequence of this as physical parts are shipped from one centre to another so the corresponding part objects must migrate from one centre database to another. Data collected from production and measurements tasks on a part will be stored in the local centre object database and forwarded to the central database. Once the detector is online it is esential for event recognition programs to have access to detector characteristics in order for event reconstruction and calibration to take place. This physical data collected during the detector assembly phase and must be arranged and processed to populate the calibration database also known as the reference database. Figure 1 illustrates the distributed database architecture across the centres.

3 Workflow Management in CRISTAL

A workflow management system (WfMS) [3] is a system that completely defines, manages and executes workflows whose order of execution is driven by a computer representation of the workflow logic. Workflows are collections of human and machine based activities (tasks) that must be coordinated to facilitate the interworking of groups of people. In CRISTAL it is the workflow system that "glues" together the different organisations, operators, processes, data and centres into a single coordinated managed production line. In general there are essentially two types of workflow that can be identified, coordination-based and production-based. Coordination based workflows are evolving workflows defined to support knowledge workers and are suited to applications which involve developing a strategic plan and responding quickly to requests. Production-based workflow is a more structured, predefined process that is governed by policy and procedure. Areas in which this type of workflow is applicable are configuration management, document routing and product life cycle management in systems manufacturing.
From the standpoint of the Coordinator CRISTAL is a production-based workflow management system in that it keeps track of and coordinates the activity

of the production of the ECAL detector. From the physicists point of view the CRISTAL system stores and manages large volumes of scientific data and provides tools to search, retrieve and analyse it. Because of the scientific nature of the application CRISTAL has the following particular characteristics that distinguish it form other production workflow management systems:-

- it is an once off production rather than a repetitive production line
- it manages large quantities of complex structured physics data
- it deals with long production time scales with very long associated transactions
- it has workflow specifications will not be fully known even at the start of production
- its workflow specifications will change during production
- it is a very distributed system

The nature and construction of CMS means that not only will the result be just one product but that this product must be complete and correct at a fixed point in time. Due to the length of time and cost of the production activities, the production process has been distributed across many centres in different countries. However, the production process must continue to run even when the connection to the CERN central system is lost. Hence the CRISTAL workflow management engine has to be designed to run at each local centre and to synchronise its activities with the central system. For the phyicists, the measurements and data taken in the production process is crucial for the analysis of the results when the experiment is assembled and run. Traditional scientific applications that involve large amounts of data have focused on the database management side of the experimental data. It is the concentration on the management and coordination of the process and the context in which the scientific data is obtained that makes the CRISTAL system different from other work flow systems. The ideas used in CRISTAL appear to be similar to a newly emerging field of Scientific Workflow systems [4].

4 Workflow System Design

The main components of a workflow management system are a workflow application programming interface and a workflow enactment service. The workflow application programming interface allows administrators to specify workflows, specify tasks and assign them to people and machines. The specification tools used with this interface must allow the possible sequencing and precedence ordering of tasks to be described to fulfil the goals of the work process. The programmers interface will also allow the defined workflow model to be analysed and simulated. The workflow enactment service consists of an execution interface and an execution service provided by a workflow engine. The workflow engine provides run time services capable of creating, managing and executing workflow instances whihc had previously been defined with the programmers

specification tools. The execution interface is seen by the end users. In the case of CRISTAL the end users or operators will be guided and prompted via this interface to perform correct sequences of procedures and measurement tasks on production parts.
The specification of workflows is done using a scripting or graphical language. It must have the expressive power to specify the order of task processing and synchronisation, task relationships with the data handled by the process and data flow. Specification methods in common use are flow graphs, transaction scripting languages, state machines and Petri nets. Petri nets [5] appear to be a suitable specification technique and were chosen in CRISTAL for the following reasons:-

- they are graphical (and hence expressive) but have formal semantics
- they have well known and documented analysis techniques for reachability and deadlock
- they have enhanced modelling features such as colour, time and hierarchy
- they are a possible future standard workflow specification method.

Petri nets place model causal relations, so that if a token appears in a place then the condition it represents will be true. So, for example, if one input place to a transition represents a crystal and a second input place represents an avalanche photo diode (APD) mounting then a token in each place models the condition "if crystal available and APD mounting = true". Transitions model the tasks or activities to be performed so that when fired they launch messages to human operators, machines or software objects to perform the task. The task may well be composed of a number of subtasks the coordination of which may be modelled as an atomic transaction so may well have the states (executing, prepared to commit, committed, aborted). Failure of tasks is a possibility in CRISTAL so when this occurs it will result in the tokens being put back to the input places. In this way the Petri net models a series of production task sequences ordered in time. The part then follows a route through the production scheme according to the firing rules.
The Workflow Management Coalition (WfMC) [3], a standards body drawn from the community of Workflow Management System (WfMS) vendors, has begun to identify the primitives from which any WfMS should be built. The CRISTAL design has attempted, where possible, to adopt its recommendations. The WfMC architecture is fast becoming a defacto industrial standard so that future WfMC-compliant WfMS products are likely to emerge and could later be incorporated via an API specification into CRISTAL. The WfMC have identified a set of six primitives with which to describe flows and hence construct a work flow specification. With these primitives it is possible to model any workflow that is likely to occur. All of these workflow primitives can be modelled by a Petri net and are shown in figure 2. The design of the workflow enactment service that is the Petri net specification execution service or work engine has posed a number of problems among them :-

- one workflow engine service will not cope with the potentially large number of requests
- the issue of how to make the system scaleable as the number of parts and centres grows
- issues with distribution
- coping with dynamic change in workflow specification
- concurrency control problems between competing parts

The design approach taken in CRISTAL is to have a workflow specification for each part type. It is estimated that there will be about 500 types of parts and 1,000,000 parts altogether in CMS ECAL. When a part is produced or registered with the system its corresponding part object is associated with a workflow engine which is the execution instance of the workflow specification for that type of part. When an operator swipes a part barcode the object identifier is used to reference the parts own work engine stored as Java byte code (see http://www.javasoft.com) at the local centre. Because the work engine contains the current Petri net markings and hence current state of the part in the production flow, by swiping the part barcode the operator is informed of the possible next tasks that can be done on the part. Although the operator interface does not look like a Petri net, the action of selecting a task has the effect of firing a transition and hence executing the task.
There are a number choices to be made when implementing this part of the system. The work engine could be a Java applet obtained from the local database which has a WWW interface and executed on the operator's Java enabled browser. The main disadvantage with this approach is that the applet will not be able to access the local host file system or invoke local operating system commands because of Java security mechanisms. If however the Java work engine is stored and interpreted locally then these access restrictions do not apply. The disadvantages of just using Java for the work engine is that it limits the client server interaction with other distributed objects. Although some of the tasks are human based and can be directed to the user interface many of the tasks are machine based and require invocation of distributed object machine interfaces. An automated measuring instruments called ACCOS is one example. This instrument, specially developed for ECAL, measures dimensions, light yield and transmission spectra of crystals and can be instructed to start, stop and produce measurements through a networked object interface. On the other hand Java has the advantage that the work engine can run as an applet using a browser and the same browser can be used by the operator to view part data stored as images or documents in an OODBMS. The implementation solution that we have adopted is to use Orbix Web from Iona Technologies[6] which allows the client Java applets to be downloaded to a web browser. The Java implemented work engines can then invoke remote CORBA [7] objects using the location and access services of the local ORB. Of course the advantage of using Java implemented work engines is that being interpreted it is possible to make changes to the Petri net workflows without

having to recompile parts of the system.

5 Handling Dynamic Change In CRISTAL

One particular difficulty that has had to be overcome is that of dynamic change to the definitions of the components that CRISTAL manages. It is envisaged that particularly in the early stage of production there will be many changes to the definitions of tasks and parts. If these definitions are kept statically in the database then problems of database dynamic schema evolution will result when definitions are changed and particularly when objects migrate from one centre to another. To circumvent the schema evolution problem we have introduced the concepts of meta-objects. Meta-objects are descriptions of objects which are managed by the database. Meta-objects are customisable without affecting the underlying database schema. The concept of meta-objects comes directly from the ideas of reflection [8], that is the ability for a system to manage information about itself and to change aspects of the implementation of the system on an object-by-object basis. Each fundamental object in CRISTAL has a meta object associated with it which can manage the ongoing and regular foreseen modifications. Meta-objects not only perform the function of handling amendments to the structure of objects but they also provide the dynamic aspects of interaction between the users and the database, known as production handlers.

Workflow definitions will also change as crystal production gets underway. Typically this will mean that a workflow will acquire new activities or tasks and existing tasks will evolve. To avoid database schema changes, workflow definitions are also associated with meta-objects. This introduces the added complication that some parts will have been produced and assembled according to one workflow whilst identical parts may have flowed though the same centres but been produced according to a different version of the workflow. Of course other identical parts may have been subjected to several version changes of the workflow whilst going through the production process. The change in workflow task sequence that comes about because of a workflow version change could have an affect on the production quality of the part. So not only are the measurements stored on the part, but also the task, its version and the corresponding workflow version. Therefore each time a part is processed by a task a new task instance is created in the database to represent this activity. This object instance associated with the part holds the results of tasks and the conditions and versions under which they were carried out. In this way it is possible to maintain an audit trail or event history of all the workflow activity.

6 Project Status

CRISTAL development was initiated in early 1996 at CERN and a prototype capture tool was developed using the OODBMS O2 and WWW to allow existing aspects of ECAL construction to be incorporated in an object database. This prototype demonstrated the use of WWW with O2 and allowed the capture of histograms holding such data as transmission spectra of crystals and can be viewed using a web browser by visiting http://hpcord02.cern.ch/cristal/main.html. The second phase of prototyping and technology evaluation was initiated in the summer of 1996 and development of this prototype is well under way and is based on the ECAL testing and construction programme in CMS. The project is following the ESA PSS-05 Software Engineering Standards. At the time of writing the final version of the User Requirements Document for the CRISTAL Prototype 2 system is complete and the Software Requirements Document almost complete. The object model is based on the Unified Method [9].

Acknowledgements

The authors take this opportunity to acknowledge the support of their home institutes and to thank all those involved in the continuing CRISTAL effort. In particular the support of P. Lecoq and J-L. Faure and the help of D. Rousset, G. Barone, G. Organtini and W. Harris is greatly appreciated.

References

[1] J-M Le Goff et al., "CRISTAL: A data capture and production management tool for the assembly and construction of the CMS ECAL detector". CMS NOTE 1996/003

[2] ESA PSS-05-02, ESA Board for Software Standarisation & Control, 1991.

[3] Workflow Management Coalition. Glossary & Terminology Document No WFMC-TC-1011, June 1996
See http://www.aiai.ed.ac.uk:80/WfMC

[4] M. Weske et al, "Scientific Workflow Management: WASA Architecture & Applications".
See http://www.math.uni-muenster.de/ dbis/Weske/Common/wasa.html

[5] J. L. Peterson, "Petri Nets", ACM Computing Surveys, 9, No 3 1977

[6] Iona Technologies Ltd, Dublin. See http://www.iona.com/

[7] OMG Publications: See http//www.omg.com/

[8] J. Peters & M. T. Ozsu, "Reflection in a Uniform Behavioural Object Model", Lecture Notes in Computer Science 823, R. A. Elmasri et al (Eds) Springer-Verlag, 1994, pages 34-45.

[9] G. Booch & Rumbaugh, "Unified Method for Object-Oriented Development ".,
Version 0.8, 1995. Available from http://www.rational.com

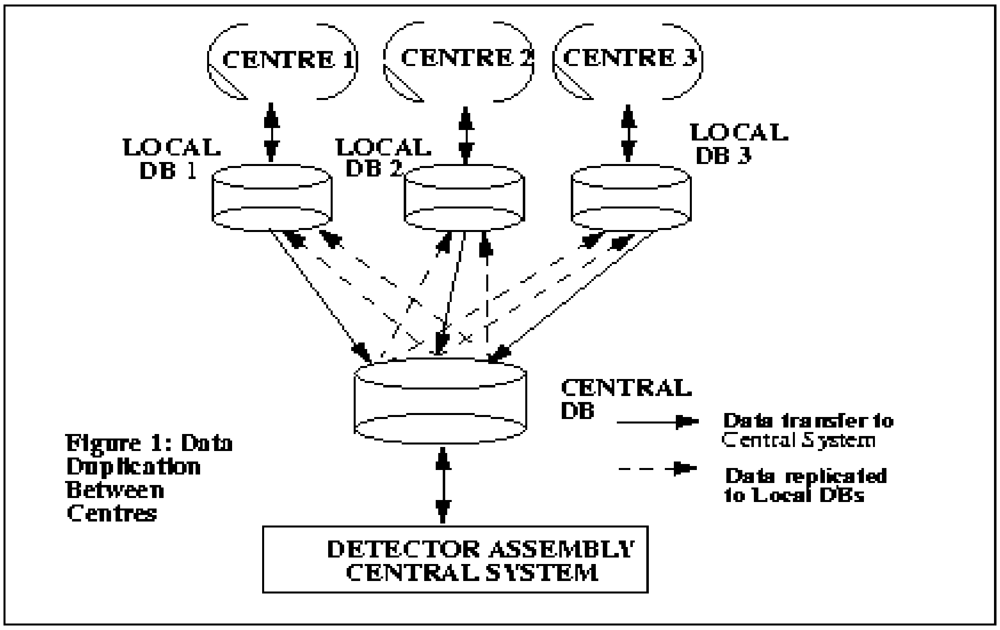

Figure 1: Data Duplication Between Centres

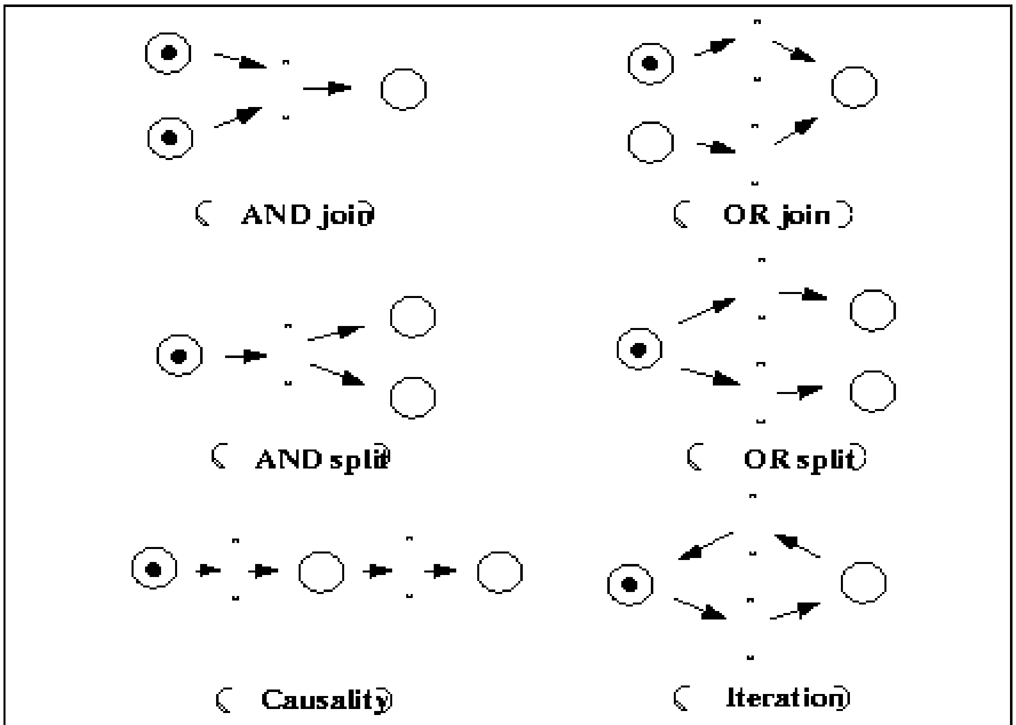

Figure 2: The six WfMC primitives