Construction Planning and Tracking for CMS HGCAL

SEPTEMBER 16, 2024

EPIC TIC MEETING

JEREMIAH MANS, UNIVERSITY OF MINNESOTA

Your Speaker

Professor of Physics at the University of Minnesota, primary research with the CMS detector at the CERN LHC

Current US role: L2 Manager (since 2017) responsible for endcap calorimeter (HGCAL) within the overall project which is delivering the US contributions to the HL-LHC CMS detector

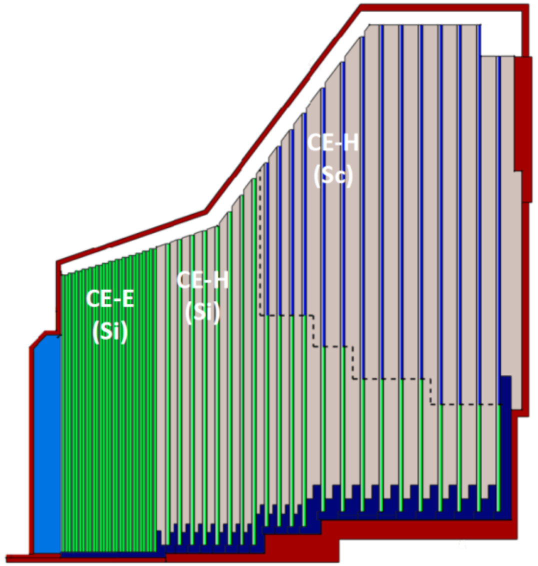
- Led the development of the US HGCAL resource-loaded P6 schedule (BoEs, risk registry, etc)
- Previously held similar role from 2015-2015 for the Phase 1 upgrade of the CMS HCAL

International roles

- International deputy upgrade coordinator 2015-2017, organized international process which led to technology selection for endcap calorimeter
- Deputy international project coordinator for HGCAL since 2022

Particular focus on the organization and coordination of the complex flows of construction required for the endcap calorimeter

A Detector of Many Parts

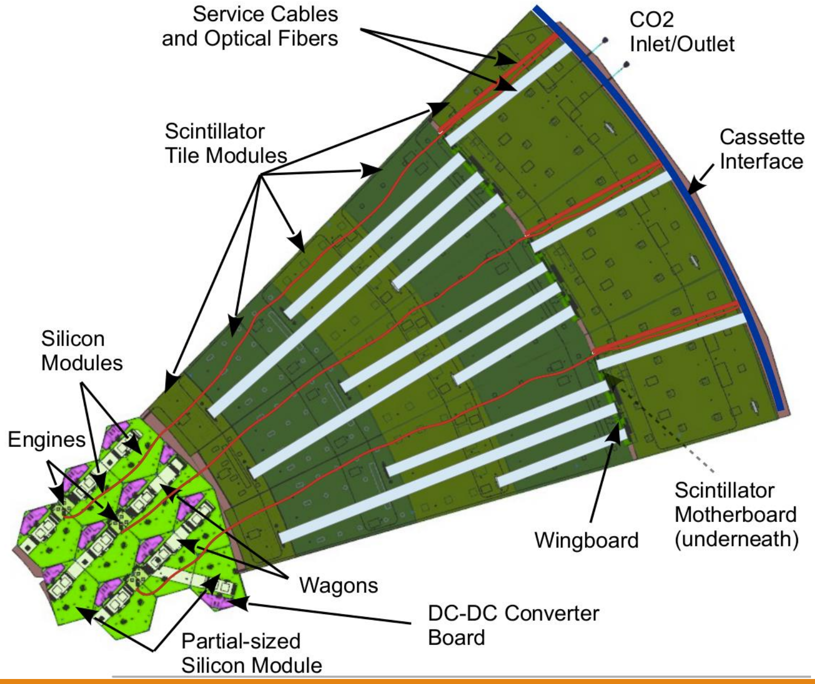


HGCAL is a sampling calorimeter designed to survive the very harsh environment of the endcap at HL-LHC

In the high-radiation environment, we use silicon sensors to construct hexagonal modules, with lead/tungsten or steel as absorber. In the lower-radiation regions, we economize by using plastic scintillator with direct readout of each tile using SiPMs

Given the endcap geometry and the international context (collaborating institutes from across the world, including large groups in the US, China, Taiwan, India, France, UK, Germany, and from CERN-as-an-institute), many different parts are needed and the construction involves complex international flows

- Considering silicon thickness and baseplate material as independent variables, HGCAL requires 22 different types of silicon modules
- Considering scintillator tile material and need to suppress some tiles at the periphery for cassette mounting hardware, HGCAL requires 30 types of scintillator modules
- LD Wagons come in 50 varieties and HD wagons in 10



Primary Software Tools in Use

Construction Flow Tool

- Custom-developed tool based on python and standard web-toolkits to simulate the construction of HGCAL and identify bottlenecks and plan responses to these issues

Progress Tracking Tool

- Visualization tool which takes results from construction flow tool to plot a baseline, a “just-in-time” schedule, and the actuals
- Used for tracking during production of the project and to help focus resources on key items as they arrive
  - Particularly helpful for long-lead-time items which start early and where there “feels” to be a lot of time in hand, even if there isn’t

Database

- Built on industry-standard tools (e.g. Oracle) using SQL and various interface tools, primarily used to record information during construction (e.g. source of actuals information for the progress tracking tool) and store quality control information gathered during construction, including per-chip configuration information which will eventually be needed for detector operation
- In general, not used for planning, but the development of the database forces a focus on parts of the project which might otherwise remain vague for long periods of time

Goals of the production flow analysis tool

The production flow analysis software is a tool to help with planning and for adjusting priorities as production proceeds

- Given the wide range of components required for the different cassettes, mapping the production order is critical to avoid having lost time during cassette assembly and final detector integration
- The tool can be used to identify, study, and ameliorate bottlenecks in the production process in advance

Strategy

- Tool developed to simulate the production of HGCAL based on a hierarchal bill of materials and a network of production sites. Simulation developed using python, using CERN Gitlab for collaborative development and web-based visualization

Typecodes

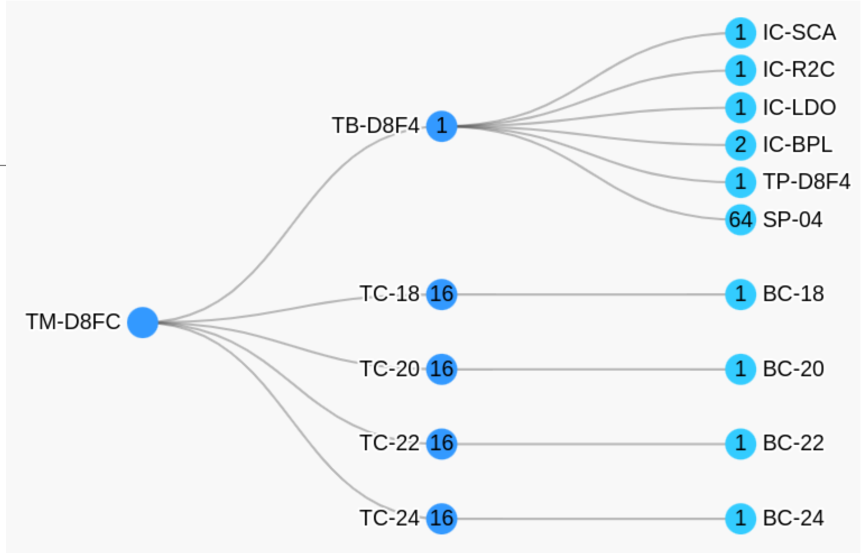
- System identifies components using the four to six character "typecode" which is part of the standard CMS labelling scheme
  - Planning tool does not track/care about serial number (N) portion of the label

320 - MM - TT [TT,TN, NN] NNNNNN

Hierarchical Bill of Materials

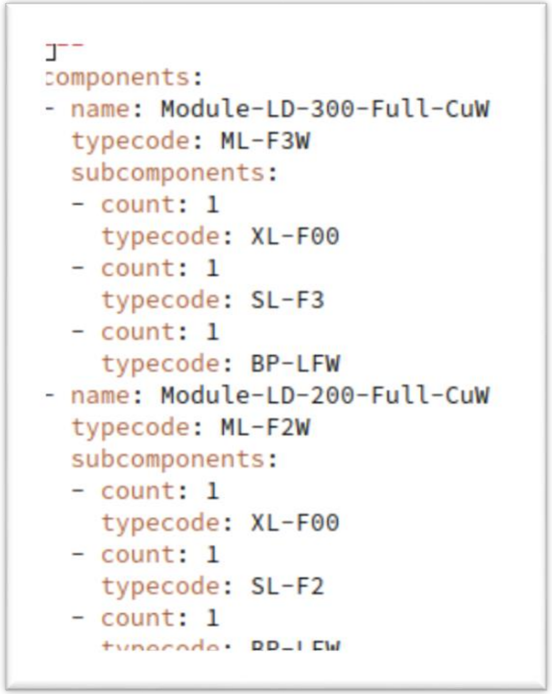
The hierarchical bill of materials contains the information about which components are required to construct higher level components

- The example visualized in the diagram shows what is required to build a D8-size tilemodule with cast tiles and 4 mm^2 SiPMs
  - Immediate inputs are tiles of four sizes and an assembled tileboard, with the assembled tileboard requiring a range of ASICs as well as 64 SiPMs



The BOM is specified using a set of YAML documents which define both leaf-node components (those with no identified inputs) and composite components which require one or more inputs for their preparation

- Components are linked by their typecode strings (**MM-TTTT**)

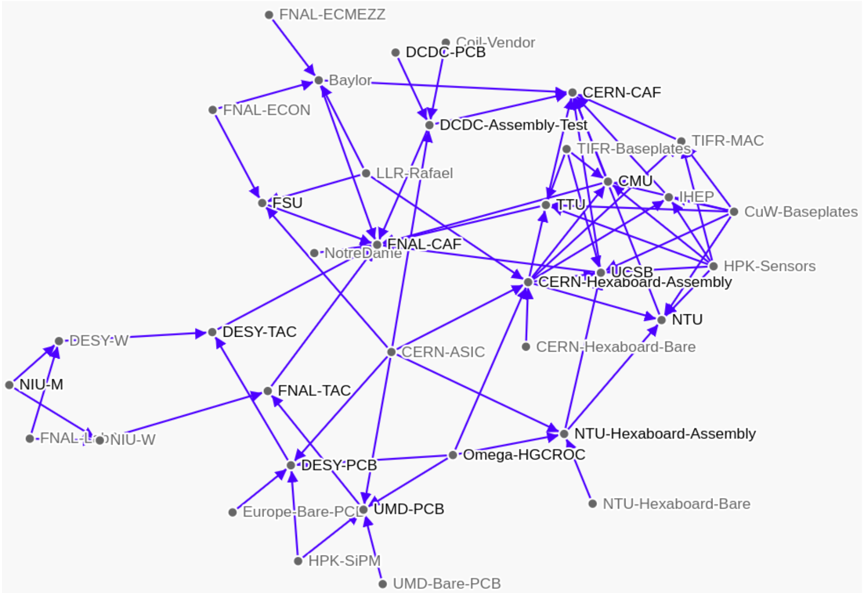


The current BOM defines 544 different components and identifies 1.7 million different items which will be constructed during production, ranging from SiPMs to full assembled endcap calorimeters

Sites

The other major component of the simulation is the sites

- Sites produce components, usually consuming other components in order to do so
- Sites are connected by transport links which hold components for some period of time representing the shipping time



The software currently define three types of sites

- Sites which produce leaf nodes on a schedule which determined externally by a vendor (e.g. silicon sensor deliveries). These sites use a simple CSV giving deliveries by date (week number) and typecode
- Sites which produce components from a list in a fixed order (e.g. cassette factories)
- Sites with a production plan which is based on upstream needs and component availability
  - Simulation optimizes production of components at each site based on overall project needs and the availability of input components

Optimized sites

Optimized sites typically produce a range of different components (e.g. silicon modules of different types)

Several parameters define the behavior

- List of components which this site will make, including the fraction of all such components which are expected to be assembled at the site
- A “dwell-time” which models the fact that sites generally cannot efficiently switch rapidly between types of components. The dwell time is the number of days in a row that the site will make the same component before considering to switch to a different component
- An average weekly production rate, which may be a constant value or which may have a ramp based on the number of components produced, representing the learning curve of pre-production and production

Based on these parameters, the currently-available input components at the site and the upstream needs, the simulation adjusts the component-under-production each time the site reaches the “dwell-time” limit or runs out of inputs

- Additional optimizations to manage corner cases of limited inputs available and other sources of inefficiency

```
sites:
  - siteid : UCSB
    name : "UCSB Module Assembly Center"
    class : Optimized
    ave_weekly_rate :
      # 2/day for first four weeks, 6/day for next 4 we
      - [ 10, 40]
      - [ 30, 160]
      - [ 60, 2000]
      - [ 90, -1]
    calendar : us
    dwell_time : 2
    produces : [ ML-T3P, ML-B3P, ML-L3P, ML-R3P, ML-T2P
  - siteid : TTU
    name : "TTU Module Assembly Center"
    class : Optimized
    ave_weekly_rate : 60
    preproduction_units : 240
    dwell_time : 3
    calendar : us
    produces : [ML-F3P*0.5, ML-F2P*0.5, ML-F2W*0.25]
```

Backwards-pass analysis

Starts with last cassettes to be produced and works backward to determine needs to meet that delivery, then moves to the next-earlier cassettes and so on

Analysis includes production rate information (including ramp), production by multiple sites, and shipping times

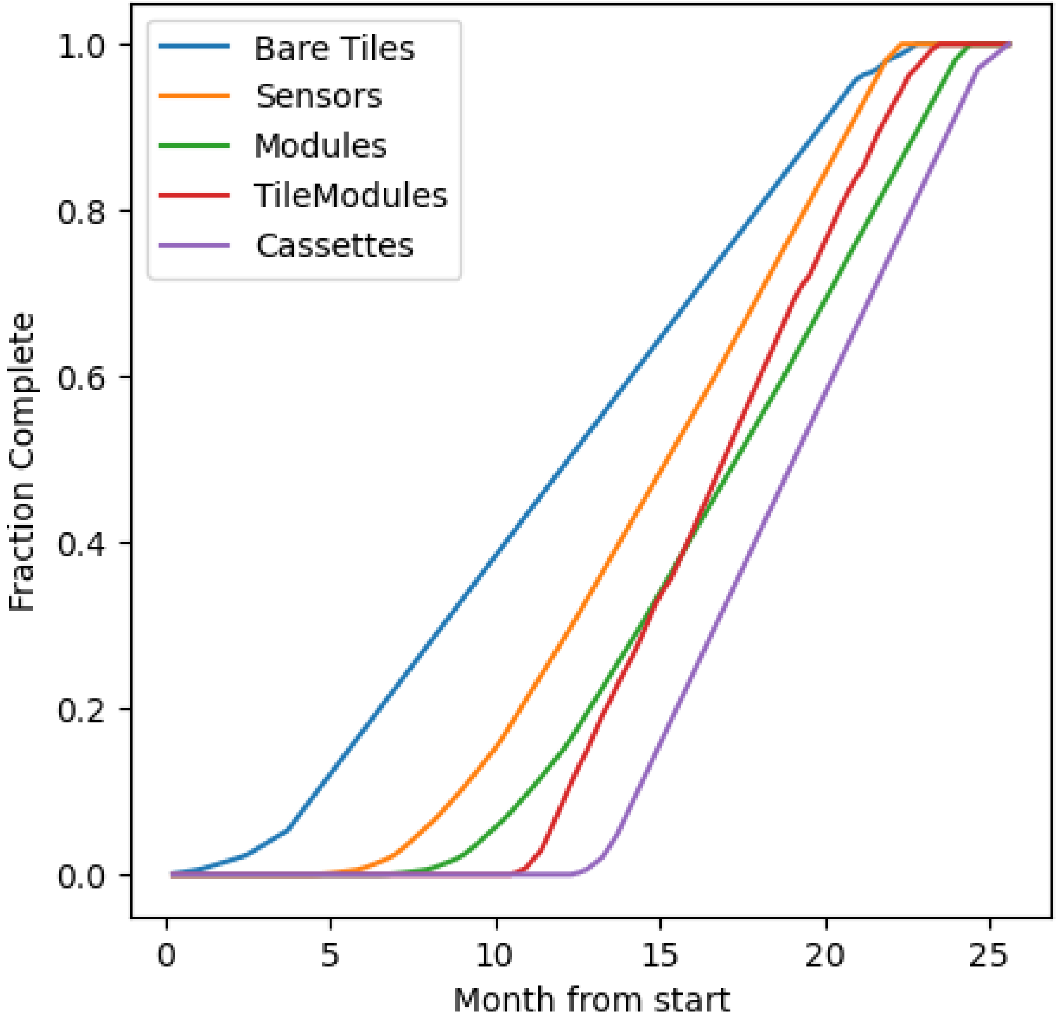
External delivery schedules, which are encoded in CSV files, are ignored and components in these files are assumed to be delivered as-needed

- Holidays are also ignored

Backwards pass provides a value for the 'fastest conceptual schedule ignoring vendor constraints'

Backwards pass results are used during forward simulation to drive decisions on what to produce at each site – logic is to always choose to produce the items which are furthest behind the ideal schedule in a global determination

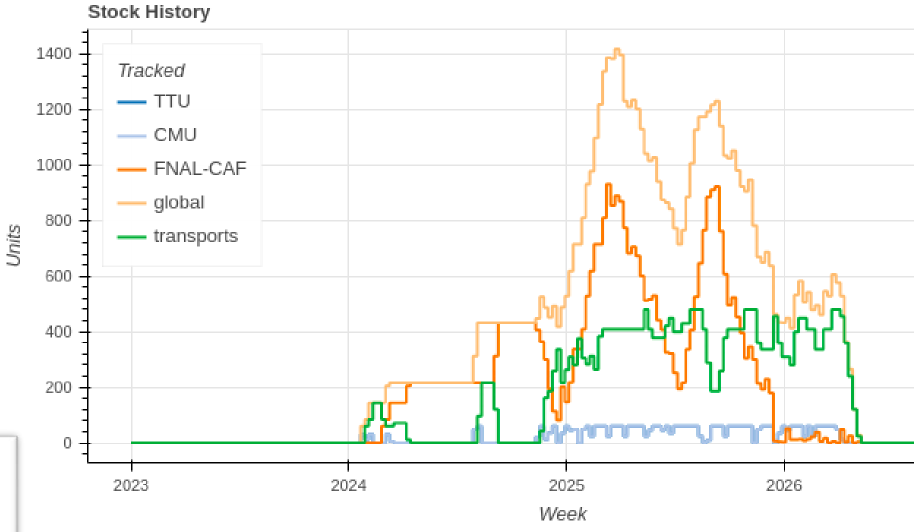
Backwards pass is **not** the baseline, as we must consider external constraints in our baseline plan



Results of simulation

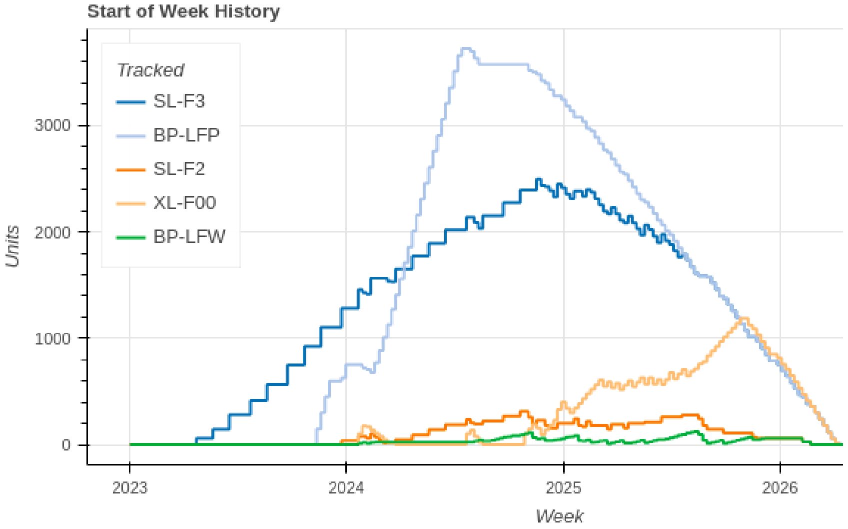
Simulation produces plots per-component and per-site

When a site is not able to produce at full-rate (e.g. a bottleneck has occurred), it can be analyzed using the Advanced Weekly Production table



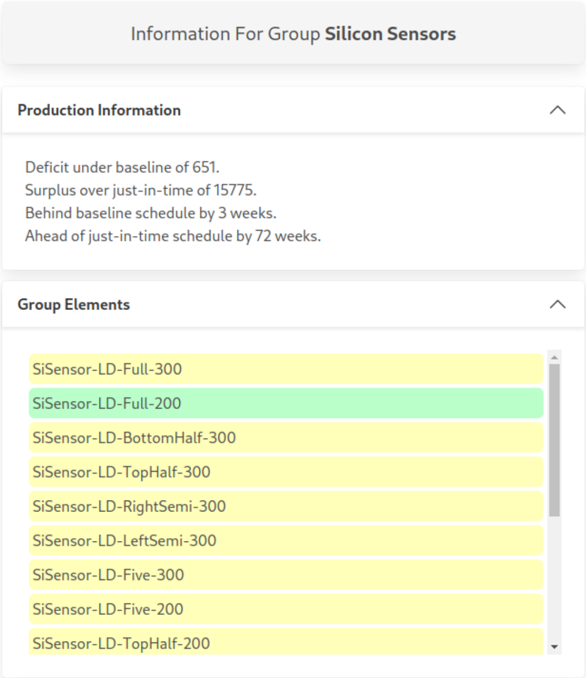
Advanced Weekly Production

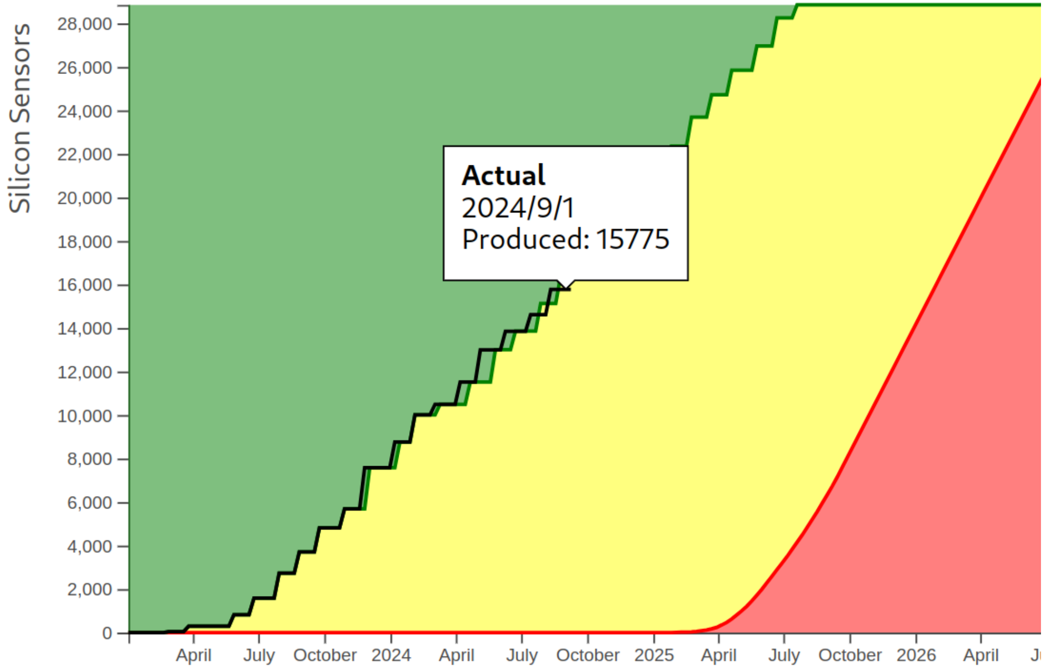
WEEK	TOTAL PRODUCED	POSSIBLE	STATUS	PRODUCTION INFORMATION
2024-11-10 (98)	40	40	OK	**Needs:** [For TM-A5FC => (TB-A5F4: 40)] [For TM-A6FC => (TB-A6F4: 40)] [For TM-C5FC => (TC-12: 272) , (TC-14: 480) , (TC-16: 480)] [For TM-D8FC => (TB-D8F4: 10) , (TC-18: 416)] [For TM-D8FI => (TB-D8F4: 16)] [For TM-K8FC => (TB-K8F9: 16) , (TC-32: 312)] [For TM-K8RC => (TC-32: 296)] [For TM-J8FC => (TB-J8F9: 40) , (TC-18: 512)] **Produced:** (TM-D8FI: 16) , (TM-K8LC: 24) ,
2024-11-17 (99)	34	40	UNDER	**Needs:** [For TM-A5FC => (TB-A5F4: 40)] [For TM-A6FC => (TB-A6F4: 40)] [For TM-C5FC => (TC-12: 129) , (TC-14: 640) , (TC-16: 640)] [For TM-D8FC => (TC-18: 416)] [For TM-K8FC => (TB-K8F9: 16) , (TC-32: 312)] [For TM-K8RC => (TC-32: 296)] [For TM-J8FC => (TB-J8F9: 40) , (TC-18: 512)] **Produced:** (TM-D8FI: 24) , (TM-C5FC: 10) ,
2024-11-24 (100)	8	40	UNDER	**Needs:** [For TM-A5FC => (TB-A5F4: 40)] [For TM-A6FC => (TB-A6F4: 40)] [For TM-C5FC => (TC-14: 640) , (TC-16: 640)] [For TM-D8FC => (TC-18: 416)] [For TM-K8FC => (TB-K8F9: 16) , (TC-32: 312)] [For TM-K8RC => (TC-32: 296)] [For TM-J8FC => (TB-J8F9: 40) , (TC-18: 640)] **Produced:** (TM-D8FC: 8) ,



Progress Visualization Tool

Based on feedback from an external industrial-efficiency institute, we developed a visualization tool which uses results from the construction flow tool to define a “baseline” (forward simulation) and “just-in-time” (backward simulation, with same end-date as forward simulation). These are plotted, along with the actual production data to produce curves indicating the progress of the project





Tool Availability

The Construction Flow Tool and Progress Visualization Tool were developed primarily by myself and a graduate student at the University of Minnesota. These tools are available under the standard Apache 2.0 license and can be used by other projects. We are available to provide limited support to particle and nuclear physics projects, but we of course have many other responsibilities

The schema for the construction database has been developed by collaborators at CERN and FNAL over period of about two decades. While the details of any license are not quite clear, in general the teams involved are open to collaboration, but very busy currently

Summary

The CMS HGCAL project is a complex one due to the endcap geometry and the goal for cost-optimization while maintaining physics potential

Several software tools are in place to plan and manage the production of the detector

- Several of these components were developed in-house as commercial software appears to typically be extremely expensive or simply unavailable
- The Oracle database is quite powerful and is used for all archival quality data. However, it is rather challenging to administer and mistakes can have large consequences, resulting in a need for conservative access rules. These aspects make the database itself less-flexible during the initial planning stage than the in-house software tools

We are still learning, as production is underway, but we are happy to share our experience and software if this will help the broader community

Additional Material

Additional details about simulation

Each site can be assigned a calendar which identifies which days are working days and which are holidays

Simulation integrates results on a weekly basis

- For a given site, calculations are done by working day, but shipments are calculated with a minimum granularity of one week
- Plots are produced on a weekly basis, starting with week 1, which was week 1 of 2023 (as sensor deliveries have already started)

Actuals

- Future planned work will allow the integration of tables of actual deliveries and production (including for Optimized sites) and carry out simulation from that point onward, but this not yet fully implemented

Physical Labels

Physical labels are produced using DataMatrix 2d bar codes which encode the fifteen character string

In addition, depending on the available space on the label, the key information from the encoded information is formatted in human-readable characters

For items to be used in the final detector, highly-robust polyimide labels designed to survive soldering cycles and low temperatures are used (e.g. Z-Supreme series)

- For shipping boxes and containers, simpler technologies can be used
- Stickers are available in a range of sizes from 8mmx8mm to 19mmx19mm and larger