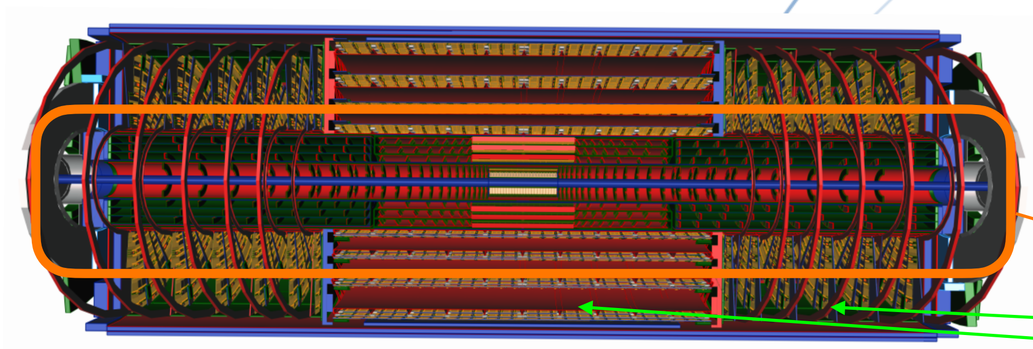
Production Modelling of the ATLAS ITk Strip Detector using the ITk Project Production Simulation (IPPS)

Fawaz Mutlaq S. Alhawiti

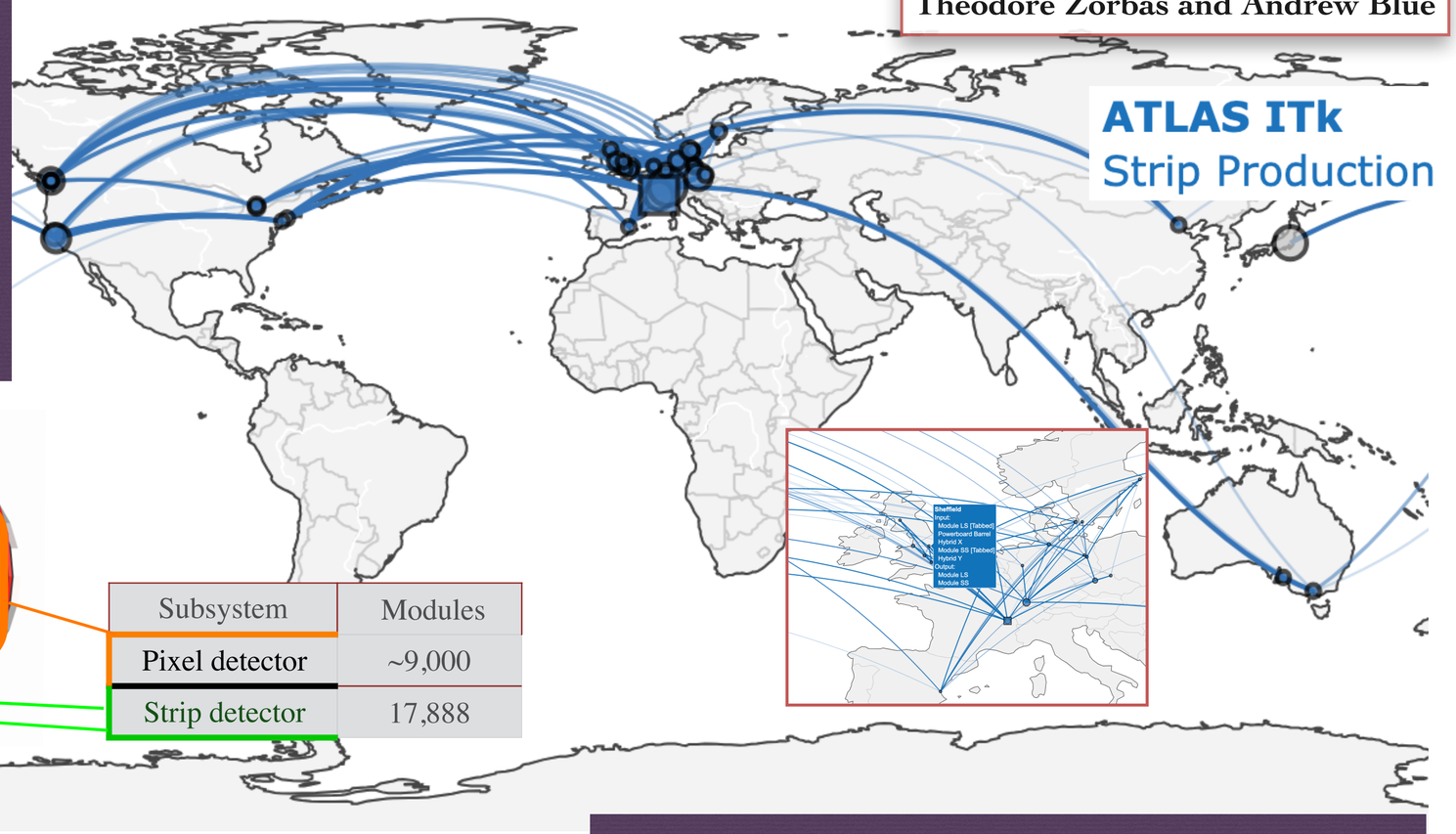
Supervisors : Kristin Lohwasser, Theodore Zorbas and Andrew Blue

University of Sheffield

Goal: IPPS is used to model the distributed production of the ATLAS ITk Strip detector, in order to predict production timelines, identify bottlenecks, and study the impact of yields, delivery schedules, and production-rate assumptions.

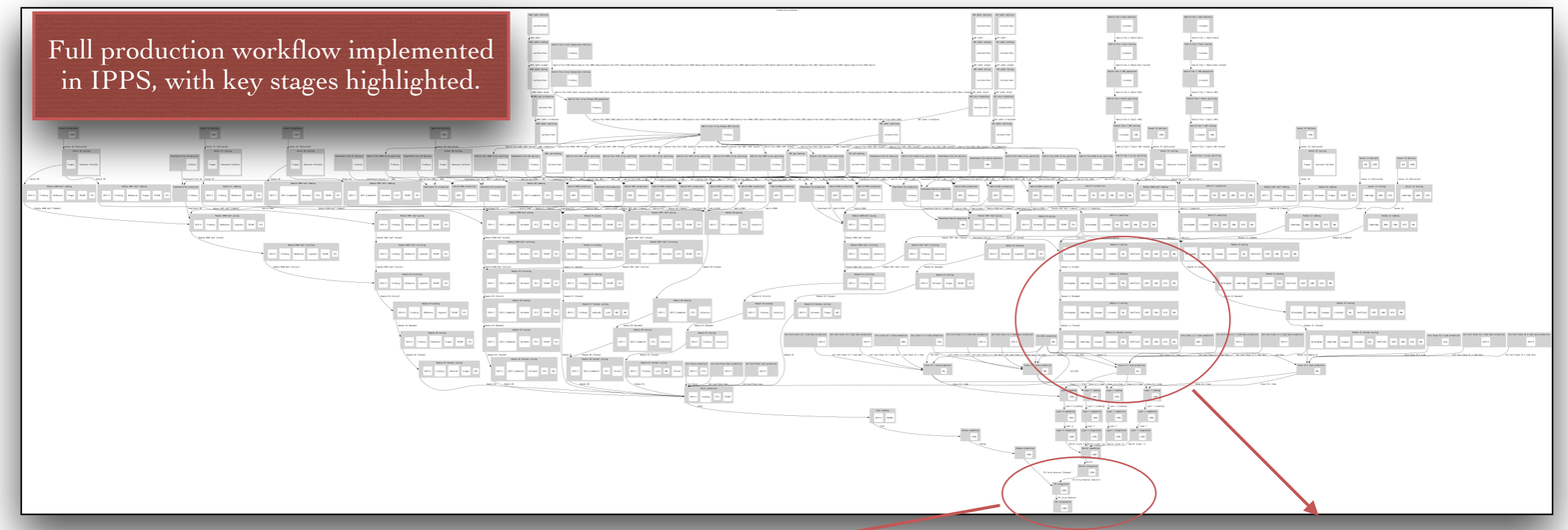


Subsystem	Modules
Pixel detector	~9,000
Strip detector	17,888

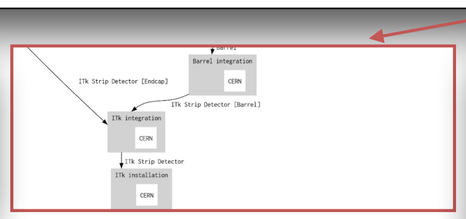


Full Process Workflow: The IPPS framework is a Python-based process-driven discrete-event simulation built using the SimPy package.

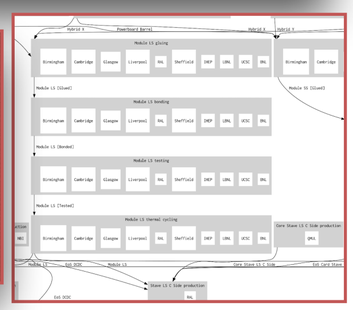
The High-Luminosity LHC (HL-LHC) upgrade of ATLAS requires a new all-silicon Inner Tracker (ITk). The ITk Strip detector will consist of thousands of silicon modules produced and assembled across multiple international institutes.



Final stage of the ATLAS ITk Strip detector integration



Example Long Strip (LS) module production chain implemented in the IPPS workflow.



Example YAML configuration describing detector components and production sites.

parts.yml

```yaml
# ------------------------------------------------------------
# Barrel: Module
# ------------------------------------------------------------

Module SS:
    Sensor SS: 1
    Powerboard Barrel: 1
    Hybrid X: 1
    Hybrid Y: 1
Module LS:
    Sensor LS: 1
    Powerboard Barrel: 1
    Hybrid X: 1

# ------------------------------------------------------------
# Barrel: Powerboard
# ------------------------------------------------------------

Powerboard Barrel:
    Powerboard Flex Barrel: 1
    AMAC: 1

# ------------------------------------------------------------
# Barrel: Hybrid
# ------------------------------------------------------------

Hybrid X:
    Hybrid Flex X: 1
    ABC: 10
    HCC: 1
Hybrid Y:
    Hybrid Flex Y: 1
    ABC: 10
    HCC: 1

# ------------------------------------------------------------
# Endcap: Integration
# ------------------------------------------------------------
Endcap:
    Disk: 6

Disk:
    Petal: 32

# ------------------------------------------------------------
# Endcap: Substructure
# ------------------------------------------------------------

Petal:
    Core Petal: 1
    EoS Card Petal Main: 1
    EoS Card Petal Seco: 1
    EoS DCDC: 2
    Module R0: 2
    Module R1: 2
    Module R2: 2
    Module R3: 2
    Module R4: 2
    Module R5: 2

# ------------------------------------------------------------
# Endcap: Module
# ------------------------------------------------------------

Module R3:
    Module R3M0 Half [Stitch]: 1
    Module R3M1 Half [Stitch]: 1
Module R4:
    Module R4M0 Half [Stitch]: 1
    Module R4M1 Half [Stitch]: 1
Module R5:
    Module R5M0 Half [Stitch]: 1
    Module R5M1 Half [Stitch]: 1

Module R0:
    Sensor R0: 1
    Powerboard R0: 1
```

sites.yml

```yaml
            Sheffield:
                - Module LS [Tabbed]
                - Module SS [Tabbed]
            RAL:
                - Core Stave LS C Side
                - Core Stave SS C Side

RAL:
    location: Oxfordshire, England
    pdb_code:
        - RAL
        #- RAL1
        #- RAL_TRANSIT
    process:
        Hybrid X production:
            commitment: 45% * 1 / 6 / 2
        Hybrid Y production:
            commitment: 45% * 1 / 6 / 2
        Hybrid X panelling:
        Hybrid Y panelling:
        Module LS gluing:
            commitment: 45% * 1 / 6
        Module LS bonding:
        Module LS testing:
        Module LS thermal cycling:
        Module SS gluing:
            commitment: 45% * 1 / 6
        Module SS bonding:
        Module SS testing:
        Module SS thermal cycling:
        Stave LS C Side production:
        Stave SS C Side production:
    input:
        Hybrid X: ((45% * 1 / 6) - (45% * 1 / 6 / 2)) * (1 + (1 / (1 + 456)))  # FIXME: addition to suppress rounding warning
        Hybrid Y: ((45% * 1 / 6) - (45% * 1 / 6 / 2)) * (1 - (1 / (1 + 160)))  # FIXME: subtract to suppress rounding warning
        Module LS: ((50%) - (45% * 1 / 6)) * (1 - (1 / (1 + 3046)))  # FIXME: subtract to suppress rounding warning
        Module SS: ((50%) - (45% * 1 / 6)) * (1 - (1 / (1 + 1618)))  # FIXME: subtract to suppress rounding warning
    shipping:
        to: CERN

Sheffield:
    location: Sheffield, England
    pdb_code: SHF
    process:
        Module LS gluing:
            commitment: 45% * 1 / 6
        Module LS bonding:
        Module LS testing:
        Module LS thermal cycling:
        Module SS gluing:
            commitment: 45% * 1 / 6
        Module SS bonding:
        Module SS testing:
        Module SS thermal cycling:
    shipping:
        to: RAL

IHEP:
    location: Beijing, China
    pdb_code: IHEP
    process:
        Hybrid X production:
            commitment: 5%
        Hybrid Y production:
            commitment: 5%
        Hybrid X panelling:
        Hybrid Y panelling:
        Module LS gluing:
            commitment: 5%
        Module LS bonding:
        Module LS testing:
        Module LS thermal cycling:
        Module SS gluing:
            commitment: 5%
        Module SS bonding:
        Module SS testing:
        Module SS thermal cycling:
    shipping:
        to: RAL
```

IPPS uses YAML configuration files to describe detector components, production processes, and distributed production sites.
These configurations are then executed within the IPPS simulation workflow.

Installed target = 100%
Required target > 100% to compensate for downstream losses