ATLAS

Status of the ATLAS Detector description

Stan Bentvelsen

Berkeley software week May 2000

XML workshops (S Goldfarb)

- Aim:
  - Get updated information on developments of the big XML-family 'out there'
  - Share experiences in use of XML for detector description
  - Share experiences in graphics with XML
- April 14th, CERN
  - 9 presentations: XML, LHCb, ATLAS, graphics
- Last Monday, Berkeley
  - 6 presentations: XML, LCD, ATLAS, graphics

AGDD vs LCD/GLAST XML

- Similarities
  - Single source geometry
  - Different views for different applications
  - Full structure in memory (different wrt LHCb)
  - 'Minimal' (need driven) model
- Differences
  - Positioning
  - Identification
  - XML material vs external material
- Plan/suggestion
  - Create a common material database
    - clear & finite task

AGDD: What do we have

- A DTD that provides
  - 'Lego' like building of generic geometries
    - very flexible to build anything
  - some DB for materials
  - some part of Identifiers
- A Generic Model
  - That parses the XML information
- Tools
  - To visualise the geometry in many ways
- Implementation files
  - rudimentary parts of detector

ATLAS

NIKHEF

Status of AGDD

- DTD: Not much change wrt previous workshop
  - still version v4
  - extend solids with 'pcon'
    - polycone along z-axis

```
<pcon nam="test" material="Air">
  <polyplane Rio_Z="0 10 0"/>
  <polyplane Rio_Z="5 20 10/>
  <polyplane Rio_Z="4 50 100/>
</pcon>
```

- Implementation AGDD
  - some progress on sub-detectors
    - muon chambers
    - tile-calorimeter
    - accordeon outline

What do we miss

- *Explicit list of requirements*
- Identification scheme
  - not complete nor approved
  - no mechanism for other schemes: readout/trigger
  - no link 'detector description classes' vs 'XML'
- Generic model
  - no 'expaned' view, propagation of rotation/translation
  - nothing about identifiers
- XML Implementation
  - symbols & symbolic arithmetic
  - 'level of detail' mechanism

ATLAS

NIKHEF

Some Specific Questions

- **Should we use XML for description of LOGICAL structure?**
  - If so, how should logical and physical descriptions be associated?
- **Is it an explicit requirement that component identifiers be generatable from the XML source?**
- **How can we assess whether physical geometry description is adequate and appropriate for simulation use cases?**
  - Must be evaluated in the context of the overall ATLAS simulation effort
- **How are physical and readout geometries related, and what is the role of XML in defining or supporting that relationship?**

Next steps in XML development

- 'Horizontal' development
  - continue with current DTD and obtain some complete detector description
- Advantages
  - complete detector
  - challenging milestone
  - move weight to client software: involve simulation & reconstruction
- Disadvantages
  - create 'slug' to improvements of model

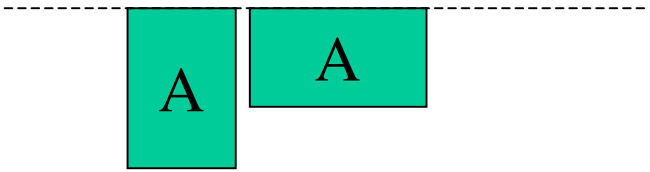
- 'Vertical' development
  - improve model more before completing description
- Advantages
  - better thought-out description
  - benefit from latest W3C developments
  - probably much easier implementation
- Disadvantages
  - no working ATLAS geometry soon
  - little development client software / generic model?

ATLAS

NIKHEF

Parameters in XML

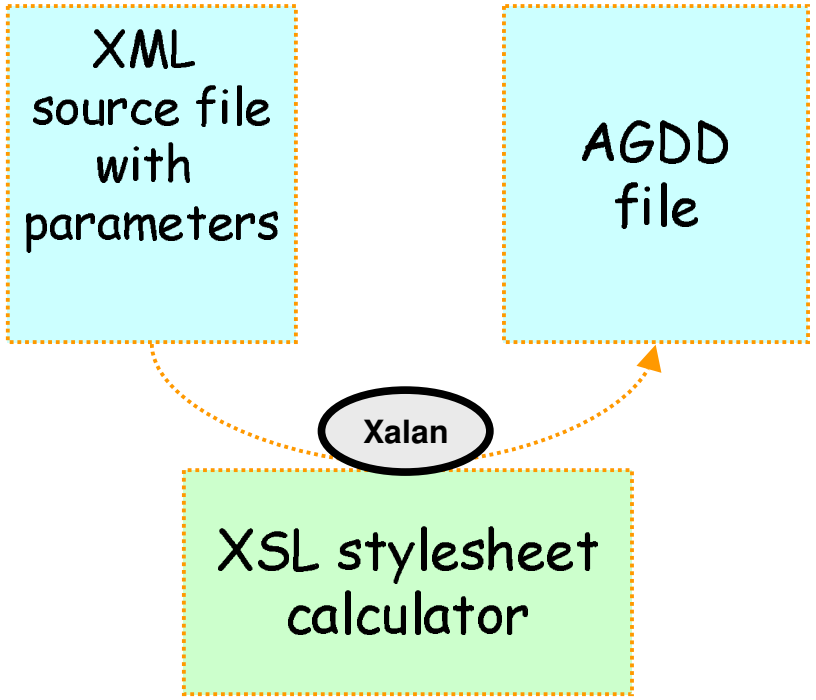
- Currently the major hick-up for implementing AGDD geometries:
  - no symbols
  - no arithmetic
- AGDD elements like 'stack' greatly reduced dependencies on numbers.
  - Still dependencies remain, e.g.



- Users requests:
  - possibility for expressions and evaluation of expressions in AGDD
  - Do we want to extend AGDD syntax to include those?
- Possibilities?
  - XSL
  - Preprocessor like 'm4'
  - MathML
  - LHCb approach
  - wait for new outside developments

Parameters in AGDD: XSL

- XSL (eXtensible Stylesheet Language)
  - Infinite more 'natural' choice on top of XML
  - possible to create a 'calculator style sheet' that resolves references?
- Needs more investigation and understanding
  - AGDD to HTML conversion no problem



I am very curious to see a working example!

Parameters in AGDD: m4

```
<!--
#
# Define the external program that performs
# the arithmetic.
#
define(calc,`esyscmd(dpcalc $1 $2 $3)')
#
# Define the parameters here
#
define(`p_boardWidth',      63.6 )
define(`p_boardLength',    128.2 )

-->
   ...

<!-- create a wafer+hybrid board -->
<box  name="SCT_board"  material="Silicon"
       X_Y_Z="1. p_boardWidth   p_boardLength"   />

   ...

<!-- create a ski of 12 boards -->
<composition name="SCT_ski">
   <mposZ   volume="SCT_board"
            ncopy="12"
            dZ="p_boardLength"
            Z0="-calc(mul,6,p_boardLength)"/>
</composition>
```

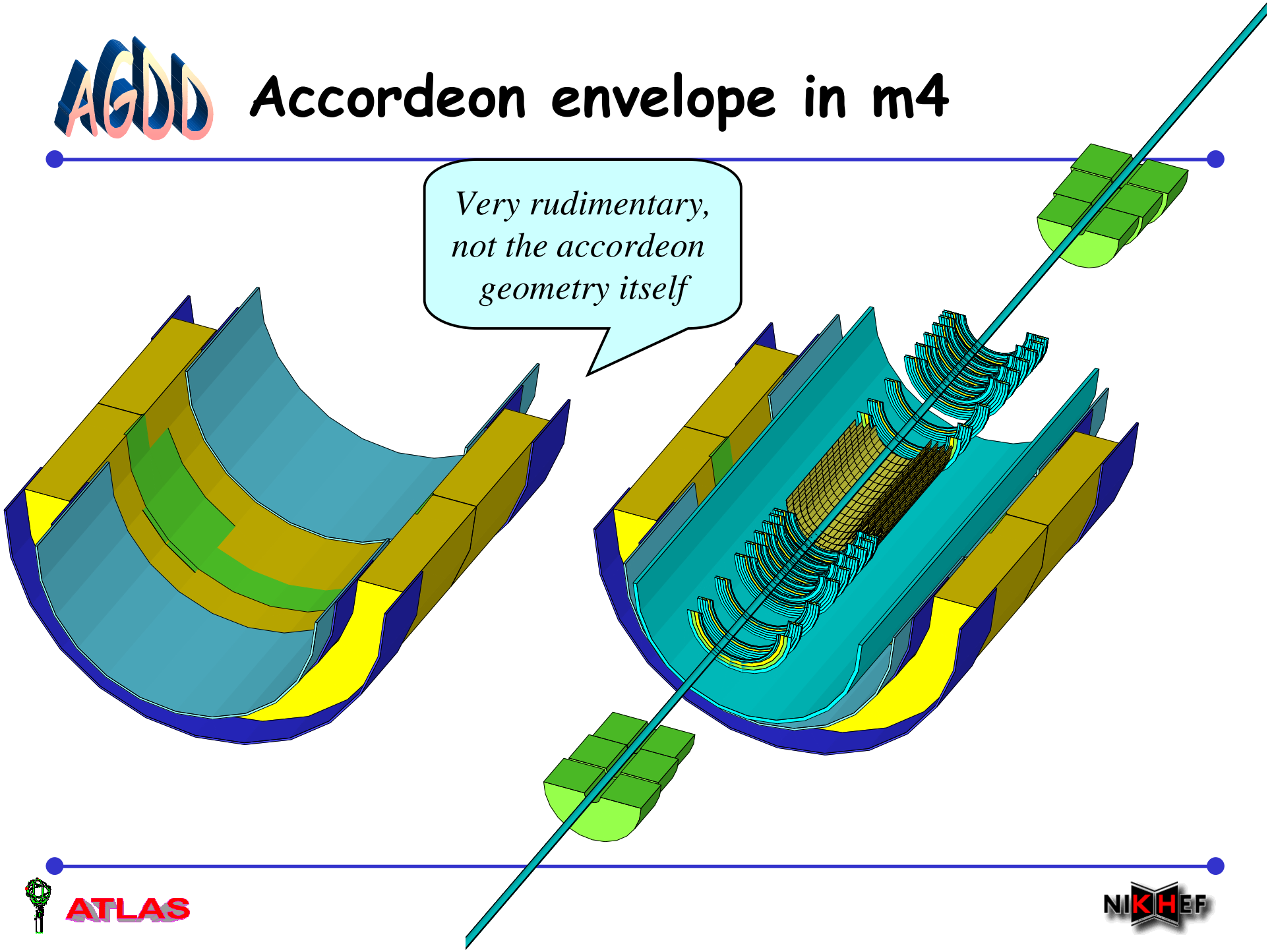
'm4' preprocessor

- define global parameters (m4-tags) in file-header
  - reference to parameters inside attribute values
- can use 'external' shell calculator to perform simple arithmetic on parameters
- preprocessed file is XML-valid
- after pre-processing using m4:
  - all parameters resolved

ATLAS

NIKHEF

Accordeon envelope in m4



LHCb solution

Numerical Expressions Parser

- Simple parser for evaluation of expressions
- Expressions can be composed of
  - integer and floating point numbers
    - 100 | 100. | .05 | 0.1| 1.34-e12|-23
  - operations: +, -, *, /, unary +|-, exponent ^
  - parenthesized expressions: 1.4 * (23.4-e12 / 1.8)
- Result is always evaluated to double value
- Operator precedence from left to right:
  - [()] [unary +|-] [^] [*|/] [+|-]
- By default checks for units in expressions

What next?

- Decide for 'horizontal' vs 'vertical approach
- horizontal:
  - bug people to get their geometry in AGDD
  - define the sub-detector envelopes
  - get Identification scheme in GM
  - 'reuse' the LHCb approach (temporarily)?
- Vertical:
  - define exact requirements
  - develop completely new alternative models:
    - e.g. try 'top-down' identifier approach in contrast to current 'bottom-up' geometry approach
  - look at the market: XML schema, MathML, etc..