Using XML to populate detector description specifications

Frank van Lingen

CMS

Eindhoven University of Technology

University of the West of England

Frank.van.Lingen@cern.ch

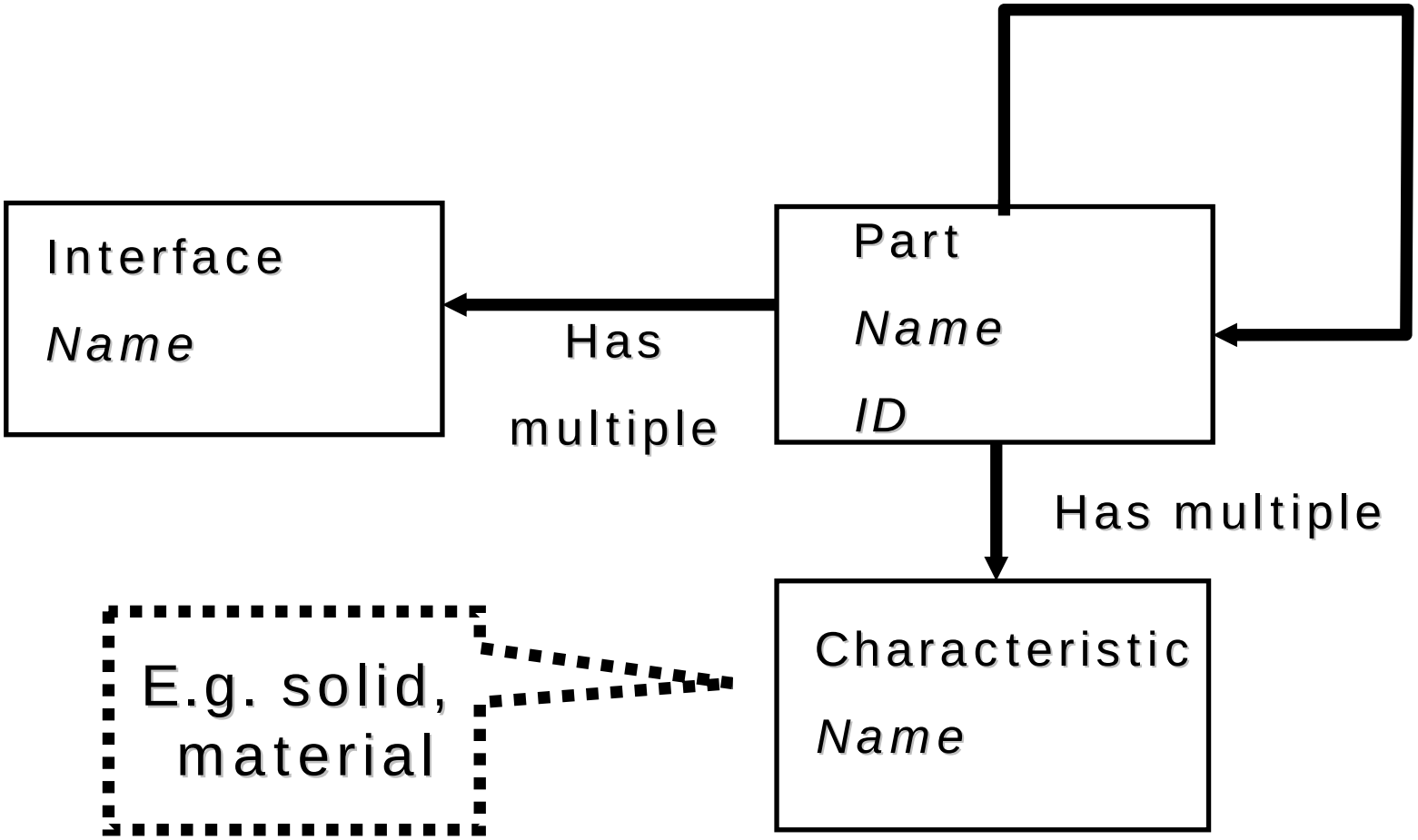
Contents

- General detector description
- Specifications (instantiation of general detector description) and detector description database
- Populating specifications
- How XML can be useful (or not)
- Conclusions

General Detector Description

- Everybody has different views on the detector ((fast) Monte Carlo, Construction, Design, TestBeam)
- Views can represent different granularities
- Abstract from these views in terms of:
  - **Interfaces**
  - **Detector parts**
  - **Detector characteristics**
- Information for detector description stored in different databases (design, construction, etc.)
- Single framework for building and populating specifications

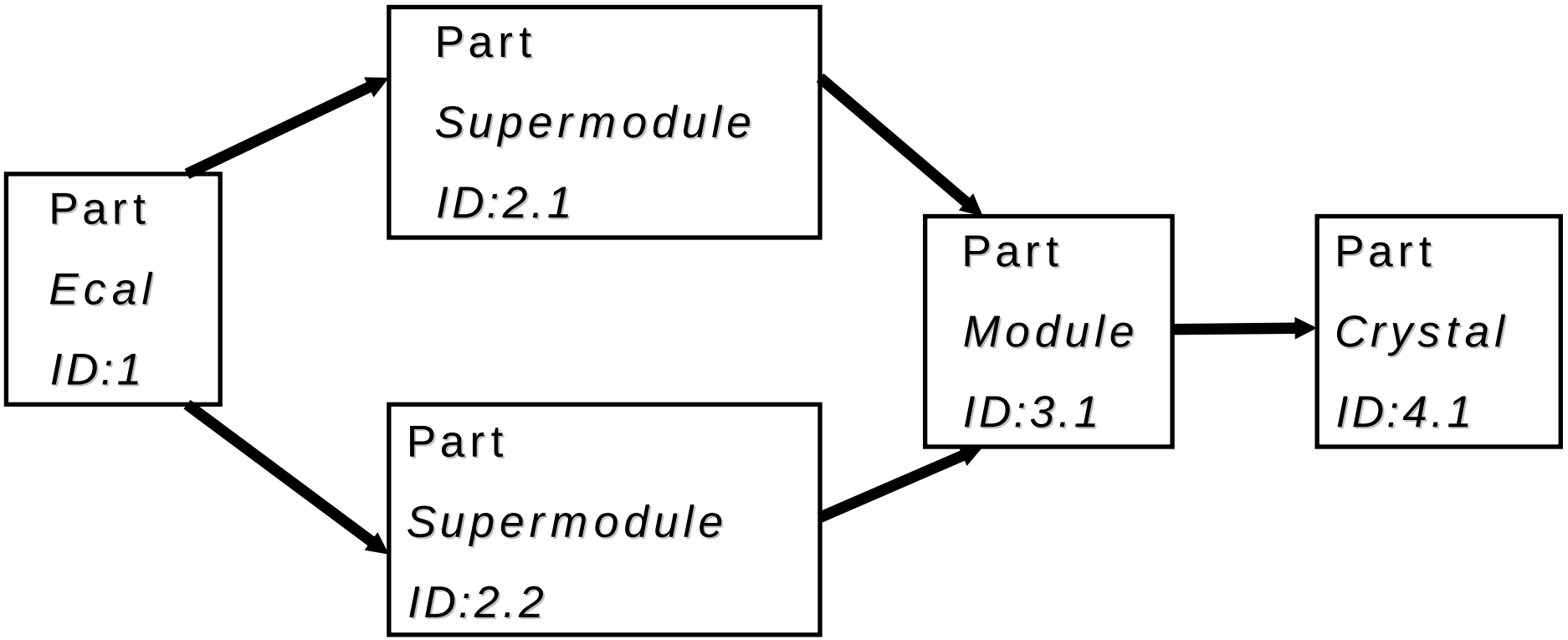
General Detector Description (simple example)



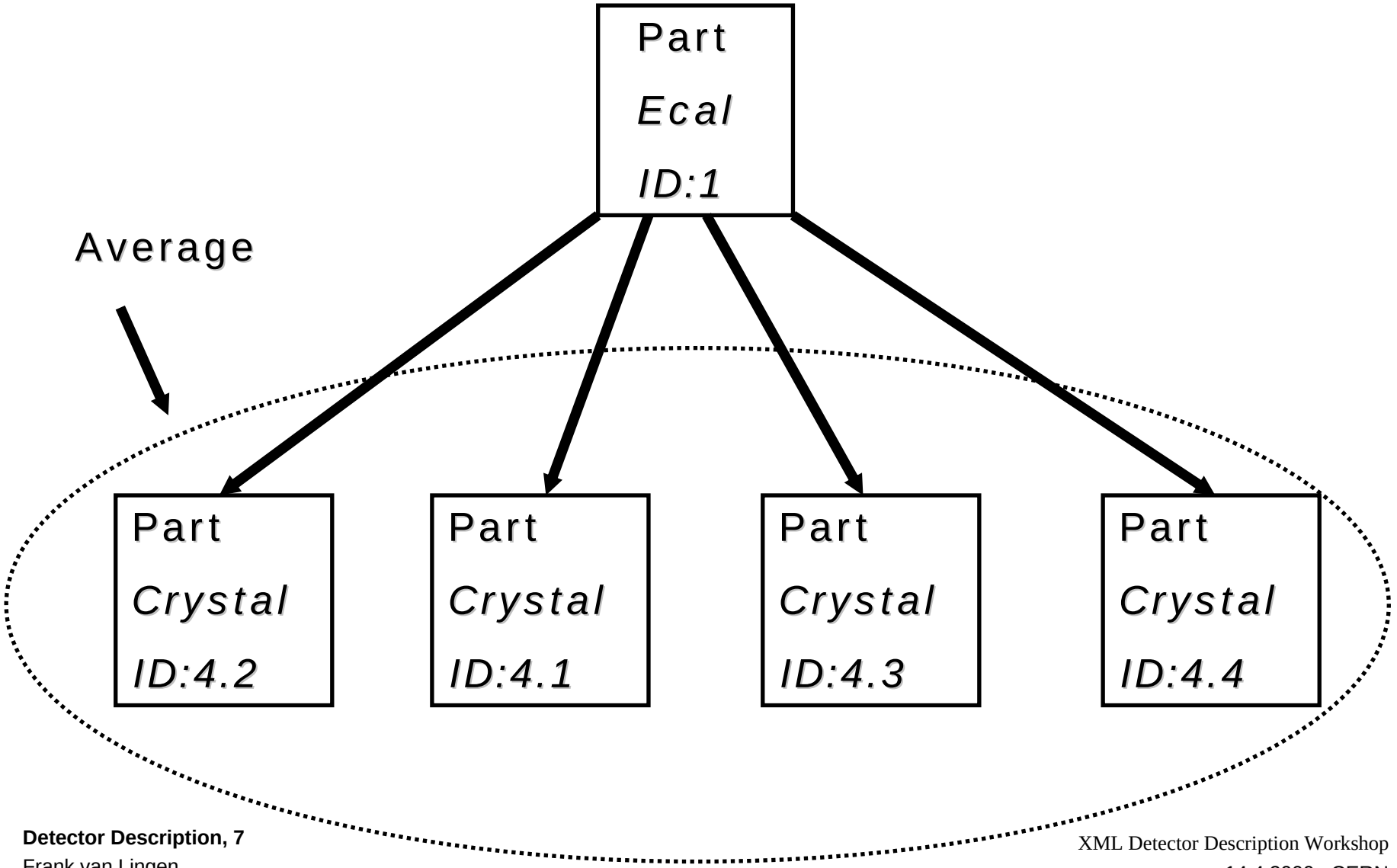
Specification

- Use general description and meta- data (specific sub detectors, parts)
- Represents a view on the detector using the general detector description
- Different applications may want to reuse parts of other specifications:
- Store specifications in a detector description database.

Specification (simple example)



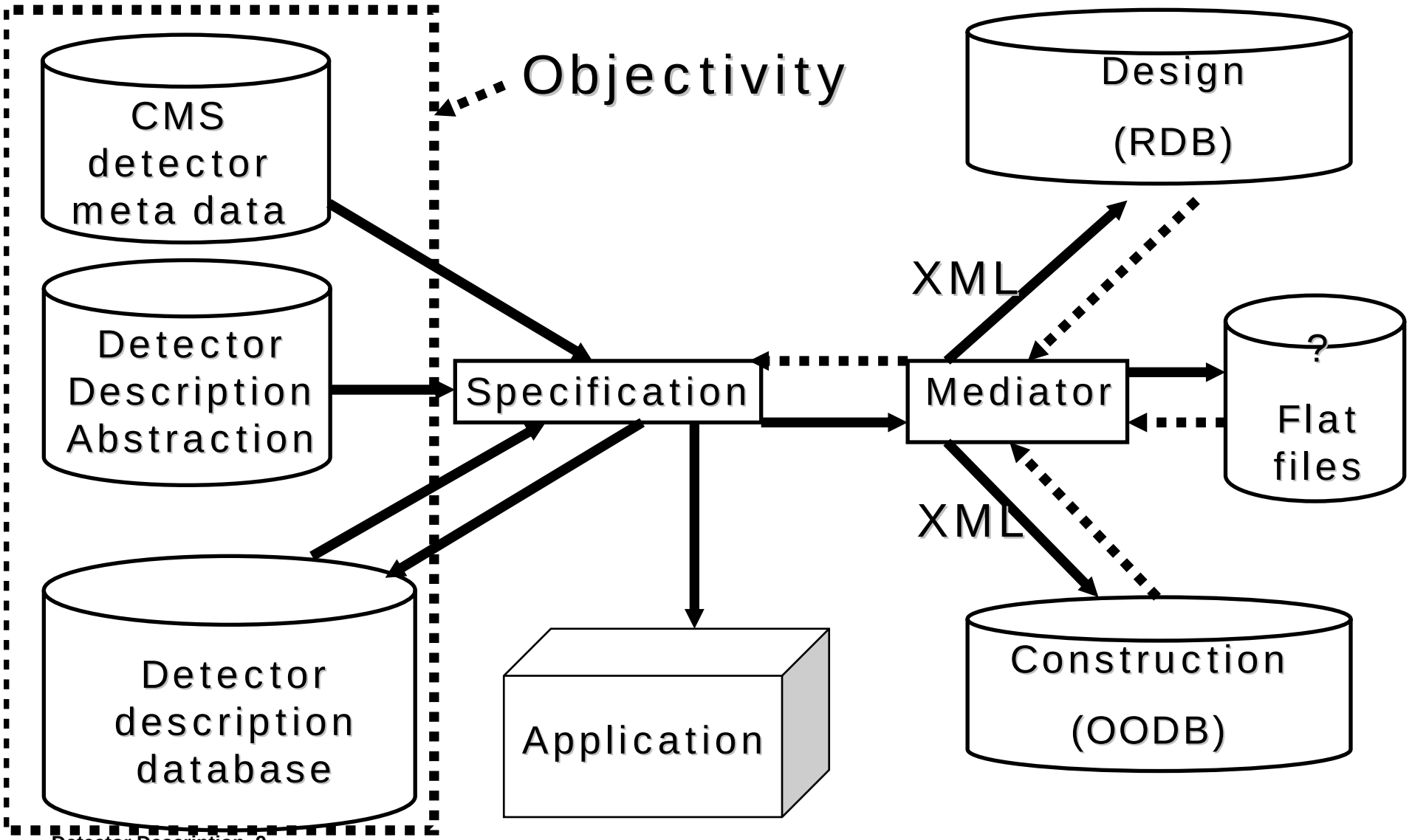
Specification (different view)



Populating Specifications

- Using "real" data sources (design, construction, control, etc.)
- Different formats have to be translated into a uniform format, using a self describing language
- Everybody uses the same "real" data on different views

Populating Specifications



Why not use XML instead of Objectivity?

- Performance (when data sets get large)
- Possible to map objects to XML but: if you use C++ applications why store in XML? You have to transform it.
- Inheritance
- Basic behavior attached to data sets
- Enables management of data (Although also possible with XML tools like: eXcelon XML Application Development Environment)
- However: not impossible with XML

Conclusions

- Usage of the same "real" data
- Definitions of different views (not forcing somebody to use another view)
- Reuse/sharing of specifications (views)
- Using XML to translate formats into uniform format
- Separating specification from data (coping with change)
- Architecture independent of CMS environment
- Maybe even detector description (largely) independent of CMS environment?
- Opportunity for different experiments to work together???

Questions?

more information about this can be found at:
http://nicewww.cern.ch/~fvlingen/research_e.html