.

Vasco Amaral
Universidade Nova de Lisboa (UNL)
vasco.amaral@di.fct.unl.pt

Overview

Overview

- Who am I?
- Projects involved
  - PHEASANT
  - BATIC^{3}S
- What do I expect from CAMPaM?

Who am I?

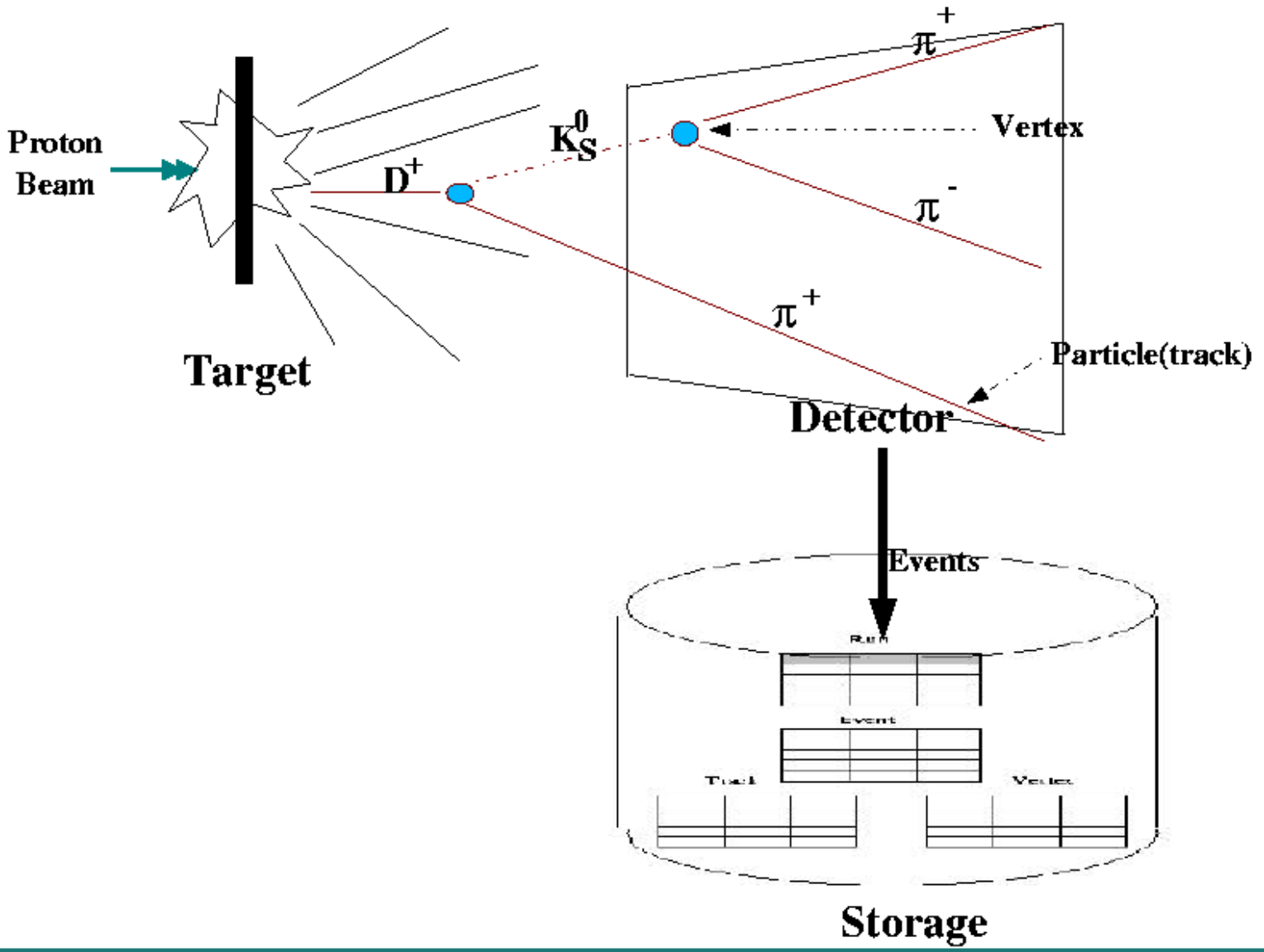
Who am I?

- 1998 Graduated in Computer Science at IST/UTL (Technical University of Lisbon)
- 1999 Worked as software engineer at CERN Geneva (Switzerland) for the ATLAS experiment.
- 2000-2003 Worked at DESY Hamburg(Germany)
- 2005 Defended Phd. at the University of Mannheim (Germany)
- Presently Assistant Professor at FCT/UNL (New University of Lisbon)

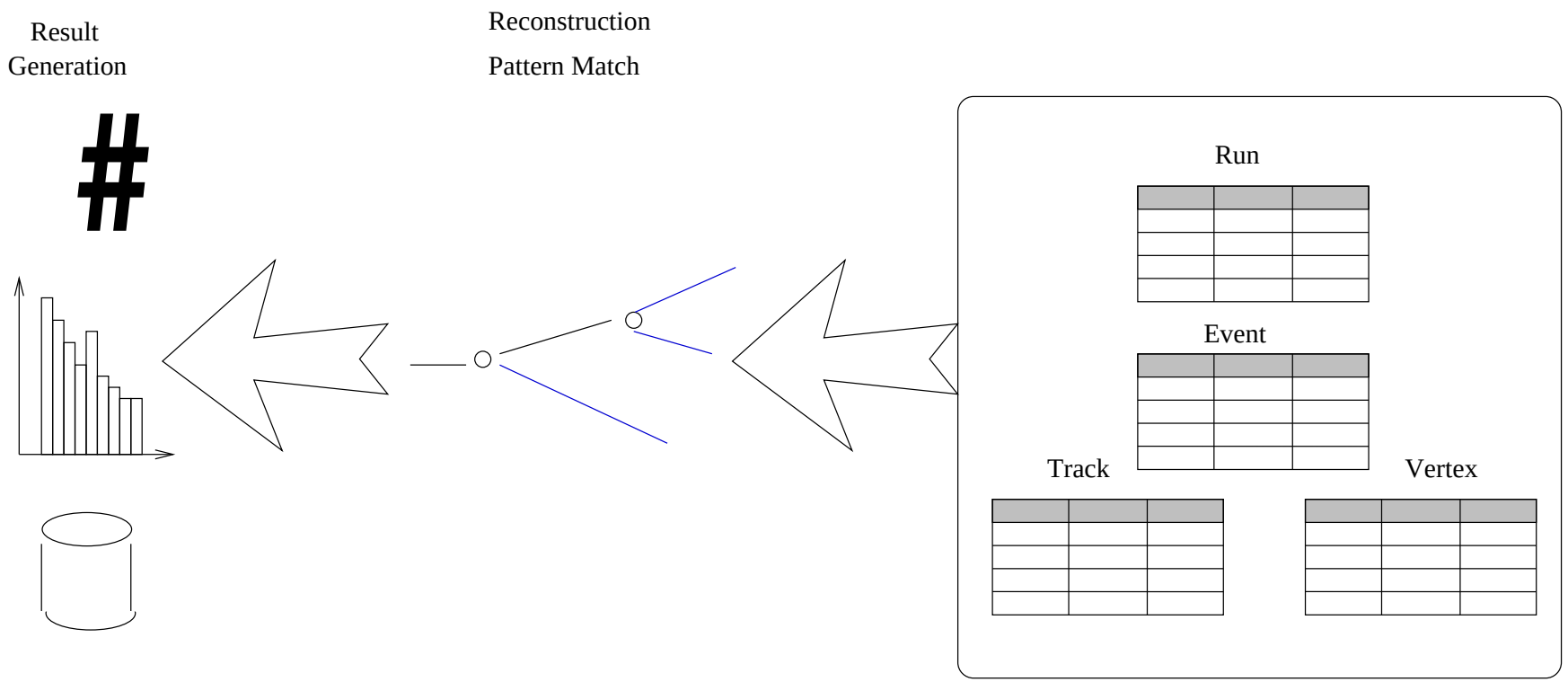
Recent interests: MDA, DSL/DSM , Model Transformation

Context: High Energy Physics(HEP)

Big and complex machines to look for fundamental particles.



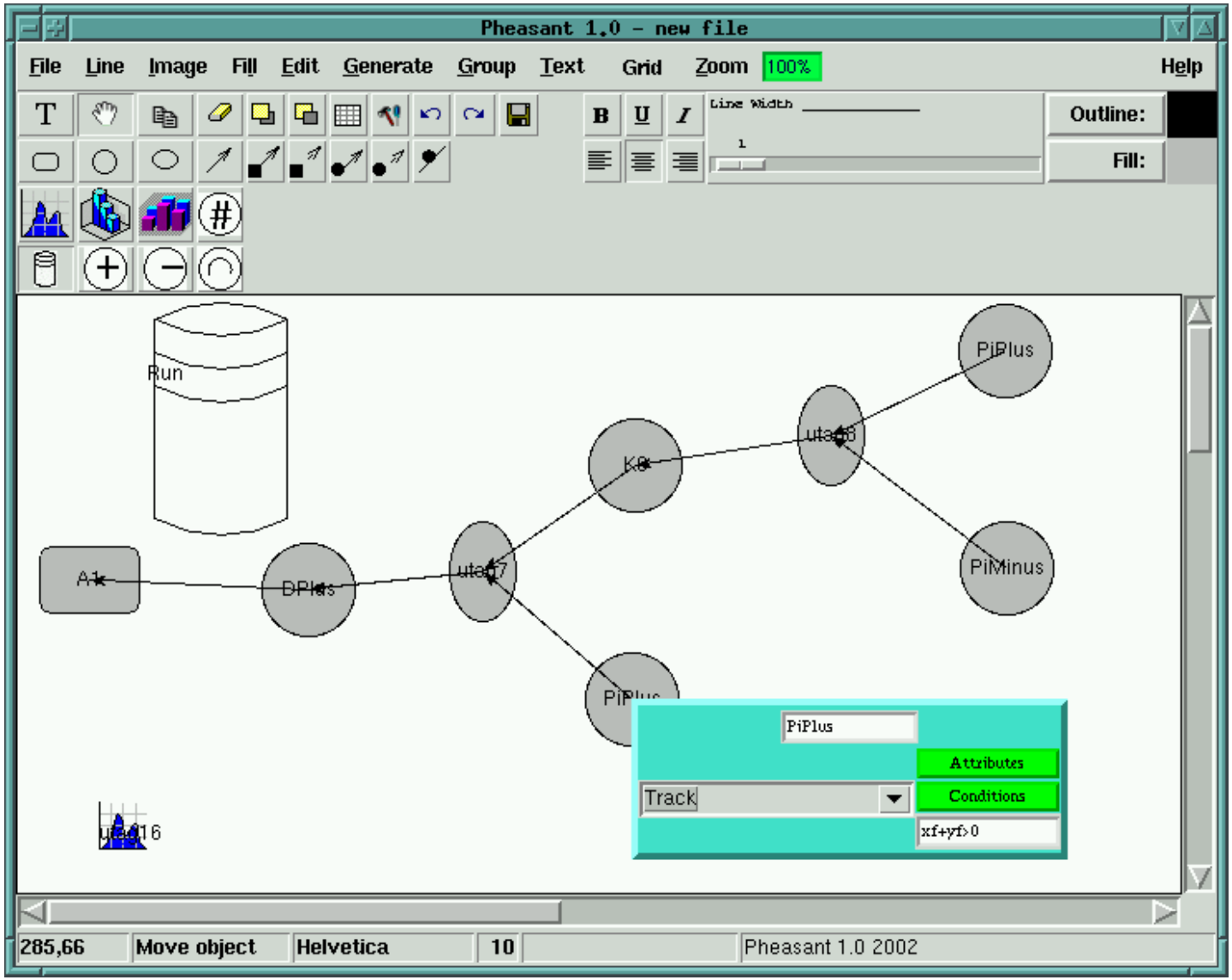
Data Mining



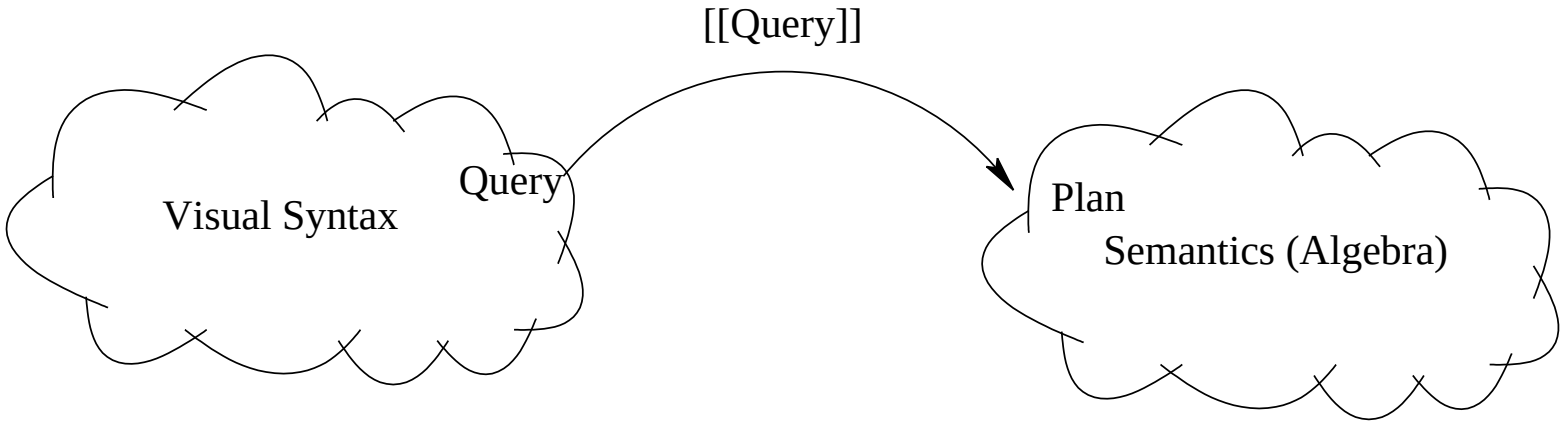
Motivation

- Problem:
  - Coding with a GPL
    Twists the way the user thinks about his data.
    Error prone (excessive work debbuging)
  - Steep learning curve for beginners (2/3):
    Demands good programming skills.
    knowing data physical/logical layout.
    Mastering the utility libraries (typically legacy systems).
- We want to increase the user productivity:
  - Getting a less steep learning curve.
  - Reduce the error rate.
  - Reduce the time spent on query generation.

PHEASANT (PHysicist's EASy ANalysis Tool)



Semantics of the language



- Visual language defined:
  Translational semantics into algebra.
  Advantage of reusing optimization techniques from the Database Management Systems area.
- Extended NF2 Algebra defined with denotational semantics.

Type System

- Basic types: Float, Bool, Integer, String
- Bulk type: $\{\tau\}$
- Tuple: $[a_1 : \tau_1, ..., a_2 : \tau_2]$
- Sub-Typing: $\tau \leq \tau' \Rightarrow \{\tau\} \leq \{\tau'\}, [a_1 : \tau_1, ..., a_n : \tau_n] \leq [a_1 : \tau'_1, ..., a_k : \tau'_k]$
- Example:

$$
\begin{aligned}
&\textbf{Event} = \\
&[id : Integer, \\
&\ particle : \{\textbf{Particle}\}, \\
&\ vertex : \{\textbf{Vertex}\}] \\
&\textbf{Particle} : [id : Integer, \\
&\ mass, x, y, z : Float, \\
&\ Energy : Float]
\end{aligned}
$$

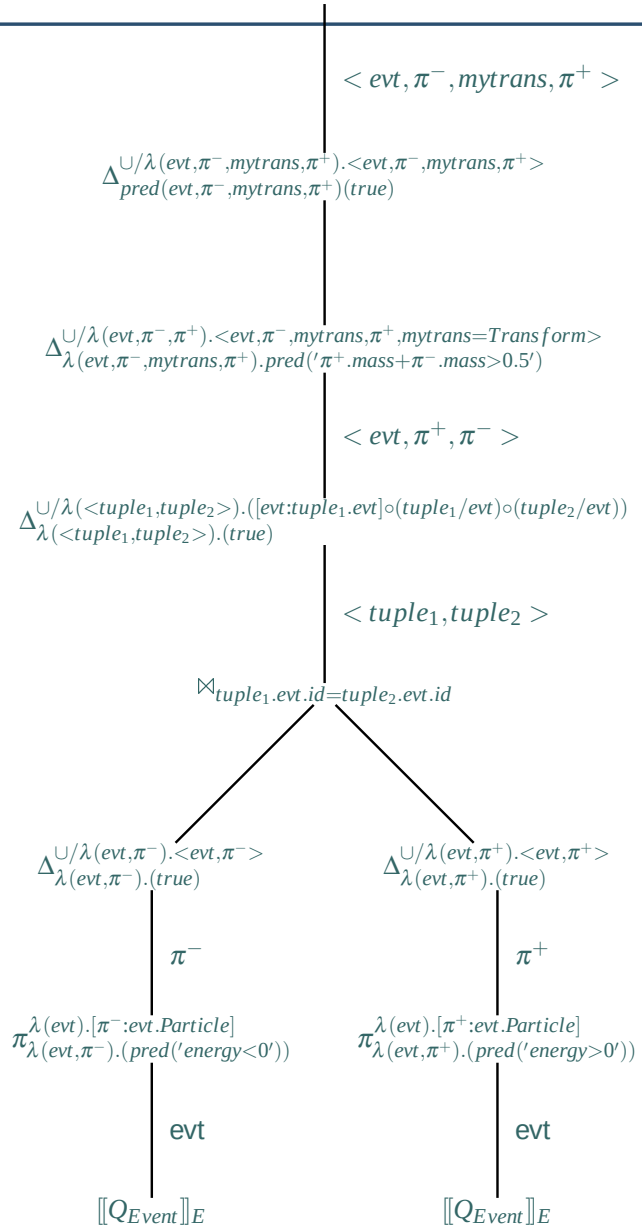
Algebraic Operators

Selection σ_{pred}	$\{\tau\} \rightarrow \{\tau\}$ $pred : \tau \rightarrow \mathscr{B}ool, \mathscr{F}(pred) \leq \mathscr{A}(\tau)$ $\tau \leq []$
Join $\bowtie_{pred}$	$\{\tau_1\} \times \{\tau_2\} \rightarrow \{[tuple_1 : \tau_1, tuple_2 : \tau_2]\}$ $pred : \tau_1, \tau_2 \rightarrow \mathscr{B}ool, \mathscr{F}(pred) \leq \mathscr{A}(\tau_1) \cup \mathscr{A}(\tau_2)$ $\tau_i \leq []$
Outer-Join $⟗_{pred}$	$\{\tau_1\} \times \{\tau_2\} \rightarrow \{[tuple_1 : \tau_1, tuple_2 : \tau_2]\}$ $pred : \tau_1, \tau_2 \rightarrow \mathscr{B}ool, \mathscr{F}(pred) \leq \mathscr{A}(\tau_1) \cup \mathscr{A}(\tau_2)$ $\tau_i \leq []$
Union $\cup$	$\{\tau\} \times \{\tau\} \rightarrow \{\tau\}$
Intersection $\cap$	$\{\tau\} \times \{\tau\} \rightarrow \{\tau\}$
Difference $\setminus$	$\{\tau\} \times \{\tau\} \rightarrow \{\tau\}$

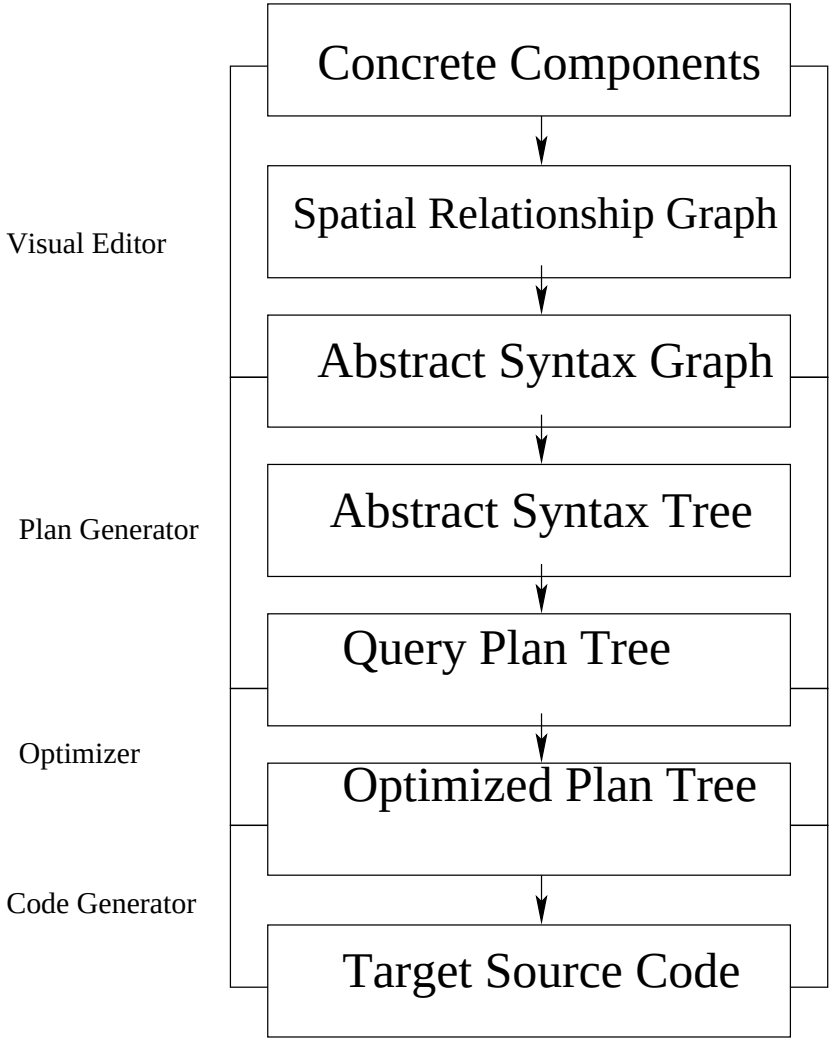
Algebraic Operators

Unnesting $\mu^{name:path}_{pred}$	$\{\tau\} \rightarrow \{\tau'\}$ $pred : \tau, \tau' \rightarrow \{\mathscr{B}ool\}$ $if\ \tau = [a_1 : \tau_1, ..., a_n : \tau_n, path : \tau_0], 0 < n, \tau_0 \leq$ $\tau' = [a_1 : \tau_1, ..., a_n : \tau_n] \circ [name : \tau_0]$ $name = \zeta()$
Outer-Unnest $=\mu^{name:path}_{pred}$	$\{\tau\} \rightarrow \{\tau'\}$ $pred : \tau, \tau' \rightarrow \mathscr{B}ool$ $if\ \tau = [a_1 : \tau_1, ..., a_n : \tau_n, path : \tau_0], 0 < n, \neg(\tau_0 \leq [])$ $\tau' = [a_1 : \tau_1, ..., a_n : \tau_n] \circ [name : \tau_0]$ $name = \zeta()$
Reduce $\Delta^{\oplus/head}_{pred}$	if $\oplus = \cup$: $\{\tau_1\} \rightarrow \{\tau_2\}$ if $\oplus = max, min, sum, ...$: $\{\tau_1\} \rightarrow \tau_2$ $head : \tau_1 \rightarrow \tau_2$ $pred : \tau_1 \rightarrow \mathscr{B}ool, \mathscr{F}(pred) \leq \mathscr{A}(\tau_1) \cup \mathscr{A}(\tau_2)$

Algebraic Operators



Overview: Query transformation process



BATICc^3s

- Collaboration with the SMV group (Geneva University), CMS experiment at CERN as Use Case.
- Build a methodology, specific to the domain of complex control systems, for specifying, building and testing 3D GUIs efficiently.

Why?

Why?

- Costly
- Difficult
- Error prone

We observe High complexity coming from :

- Number of components
- Hierarchical interaction between them
- Large number of parameters to be controled at the same time

Goal

Goal

- Specify system without the need of understand programming logic:
  - Structure and behaviour of the system and its GUI;
  - User profiles and task models;
- Translate this specification to a model:
  - Executable;
  - Verifiable;
  - Derive tests for it;
- Automatically generate a prototype.

Model Languages

Model Languages

- **Domain model** (structure and behaviour between system components)
- **Behaviour model** (component relationship with method calls and events)
- **Tasks model** (sequences of operations to achieve a goal)
- **Users model** (diferent user profiles might imply different tasks)
- **3D geometry model**
- **Presentation model** (means of interaction of GUI objects)
- **Dialog model** (associates presentation model with Users model)

Base Modelig formalism

Base modeling formalism Modeling formalism - CO-OPN based on Petri nets and algebraic data types:

- System level, which models the system behaviour and structure;
- GUI logic level, which models the semantics of operation of the GUI;
- GUI visual level, which models the presentation of the GUI.

- Adequate techniques/Formalisms for specifying DSM/DSL semantics.
- Learn state of the art approaches.
  - DS(V)L generators and Meta-Modeling tools.
  - Model transformation techniques and frameworks.
- Multi-formalism modeling (rel. multi-view).