SDMv2 Status

CALIM-2011, Manchester **Jul., 2011**

F. Viallefond.

Laboratoire d'Étude du Rayonnement et de la Matière en Astrophysique

Outline

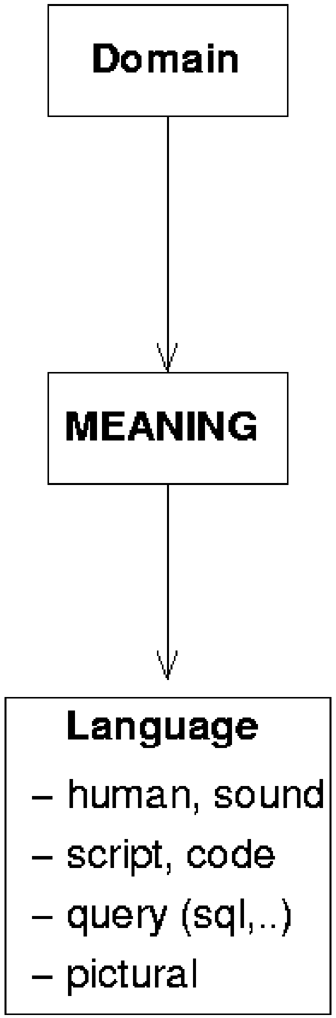
1. What is a Data Model?
    - Domain $\longrightarrow$ Language
    - Structures
    - Data Model
2. Formalization
    - Categories, Functors
    - Sketches, Models and Theories
3. Examples of diagrams
    - Primitive Data Types
    - Initial and Final objects
    - Products, Coproducts, Direct limits
4. Relation, Recursive constructs, topology of the MSDM (SDMv2)

5. Monads, lambda calculus

6. Type specification: the measurements and their topology

7. Conclusions and perspectives

Domain and Language



Sequentially organized

any languages
music

Spatially organized

graph
diagrams
symbols

Structured

grammar, rules

Primitives

words
composed words (expression)
separators (binary)

Self described

any human language
UML, XML, ...

An information system

to capture and keep memory
to exchange between systems

To represent a set of measurements

Examples of words *(physical quantities)*:

- Length, Area, Angle, Solid angle, Aperture efficiency, Rotation measure
- Speed
- Angular rate
- Noise equivalent power
- FluxDensity *(Jy which is not SI...)*
- ...

Note that:

1. All these have units.
2. Dimensioned, dimensionless and mixed case units!
3. They may have units which uses powers of rational numbers!
4. Physical expressions are composition of such words

Measurements in context

We assign domain specific meaning to words:

- Station
- Antenna
- Spectral window
- Feed
- Configuration description
- ...

Note that:

- We conceptualize
- We compose
- There are context-independencies and context dependencies
- ...

Motivations to have a data model

A measurement set is a set of concrete concepts at different levels,
a) words such as the physical quantities (universal concepts),
b) compositions of words giving rise to relations.

We must all share a common understanding of these concepts.

These must be easily usable in information systems (data reduction packages, DBMS, ...).

These must be *a)* concisely described to insure reliability and *b)* properly understood.
Very important for optimization of calculus
(architecture: structure, factorization, localization, slicing, ... i.e. geometry),

The measurement set to become a data model

The mathematicians
1/ have developed all the abstract concepts that we use (often implicitly...)!
2/ give a methodology defining what is a model and a theory!

The theory of categories: used in fundamental computer science.

Implementation of the SDMv2 is based on generic programming techniques.
The work of conceptualization which was required fits (to my surprise) very well with that theory!

Status and prospect: The 'SDMv2' will be fully explained by a theory.
From 2008 to now...

1. 2008: devel concepts of phys. quan. and several important structures, proof that it was doable demonstrated by a small proto
2. 2009: more generics (EMBRACE context), radiotelescope generic
3. 2010/11 mathematics (very steep learning curve...), converter ASDM to SDMv2
4. 2011 'reverse engineering' to elaborate the theory, more algebraic types in the implementation.

What is a data model?

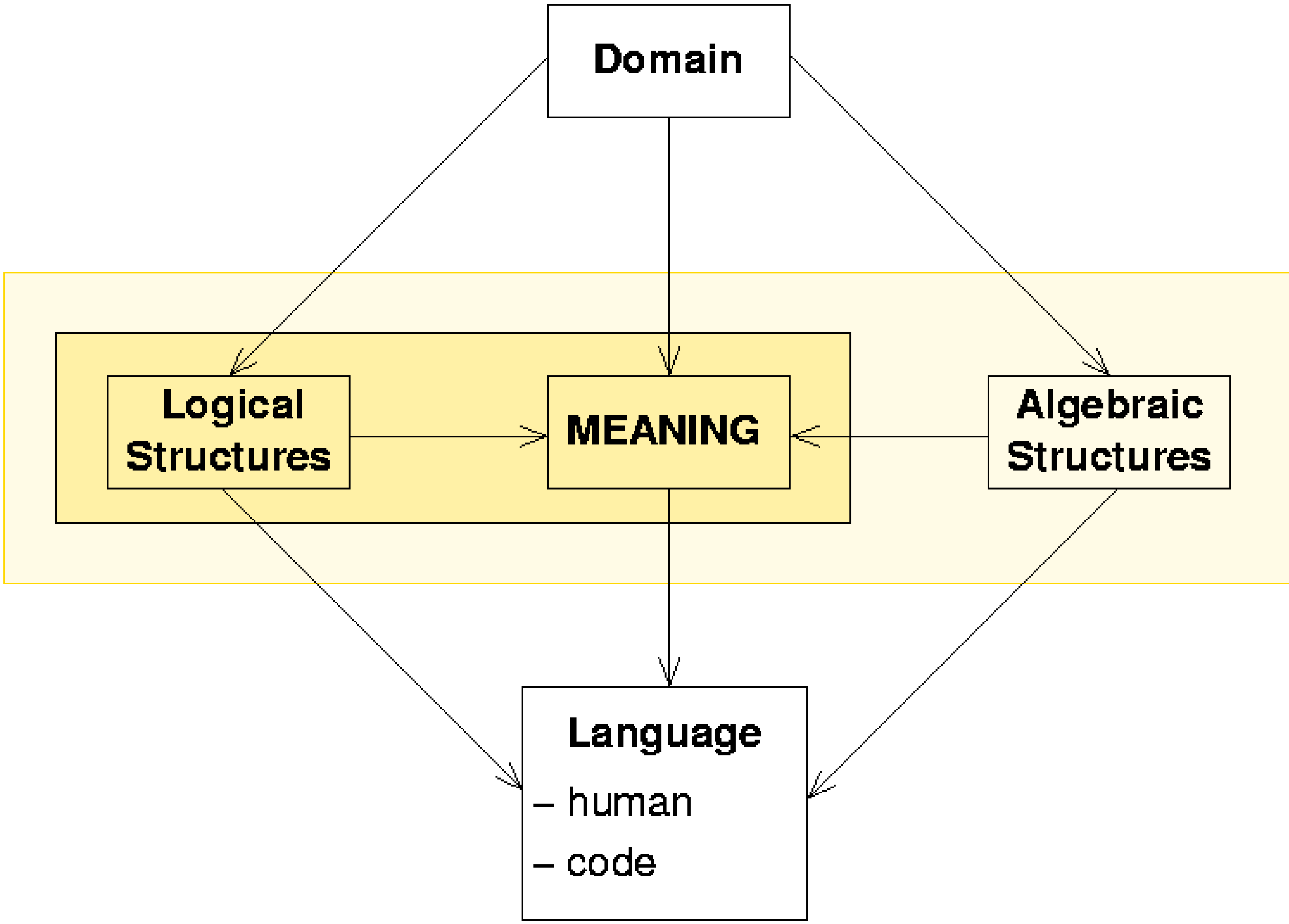
A model is the composition of
a structure (mathematical logic) with algebra.

Example: the relational data model.

- The semantic is captured through constraints.

- The structure gives the meaning of things in a formal language.

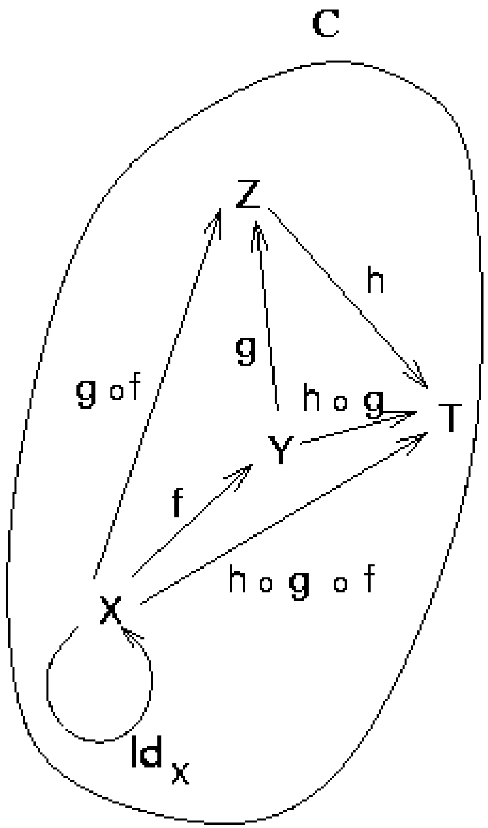
4 commutable triangles



Formalization

- Category
- Functor
- Natural transform
- Product and coproduct:
  example of diagrams, a cone and a cocone
- Direct limit
- Sketches, Models and Theories

Category
$\mathbf{C}$



Collection of objects:

X, Y, Z, T

Morphisms of objects:

f, g, h

- Identity:

$$\forall\ X \in \mathbf{C}\ \exists\ Id_X \in \mathbf{C}$$

- Transitive composition:

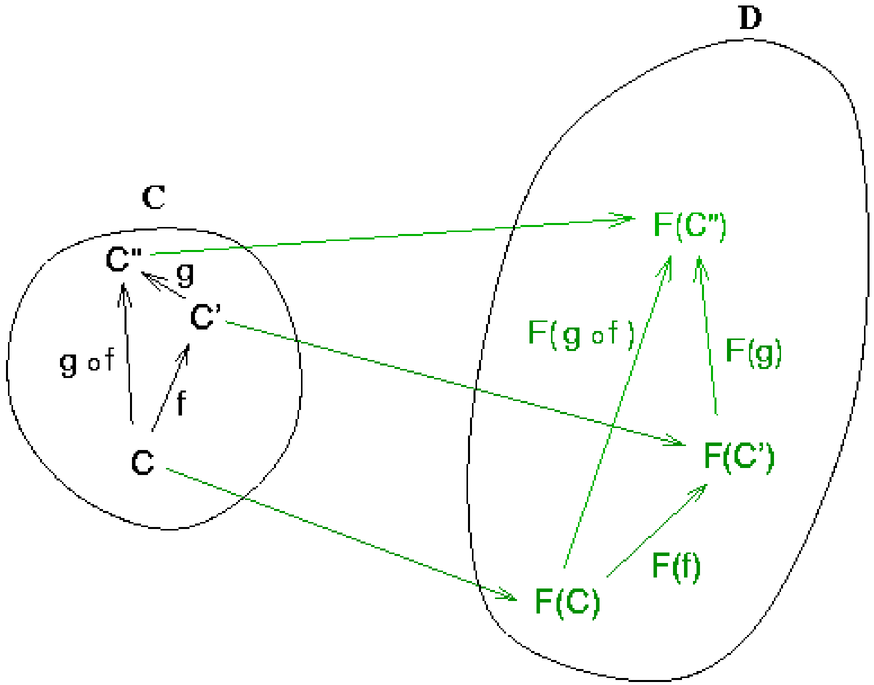
$$\begin{tikzcd} X \arrow[r, "f"] \arrow[rr, bend right, "g \circ f"'] & Y \arrow[r, "g"] & Z \end{tikzcd}$$

- Associativity:

$$\begin{tikzcd} X \arrow[rr, "(h \circ g) \circ f = h \circ (g \circ f)"] \arrow[d, "f"'] \arrow[drr, "g \circ f"] & & Z \\ Y \arrow[rr, "g"'] \arrow[urr, "h \circ g"'] & & T \arrow[u, "h"'] \end{tikzcd}$$

Functor

$F: \mathbf{C} \longrightarrow \mathbf{D}$



Two categories $\mathbf{C}$ and $\mathbf{D}$

The morphism $F: \mathbf{C} \longrightarrow \mathbf{D}$
is a functor if:

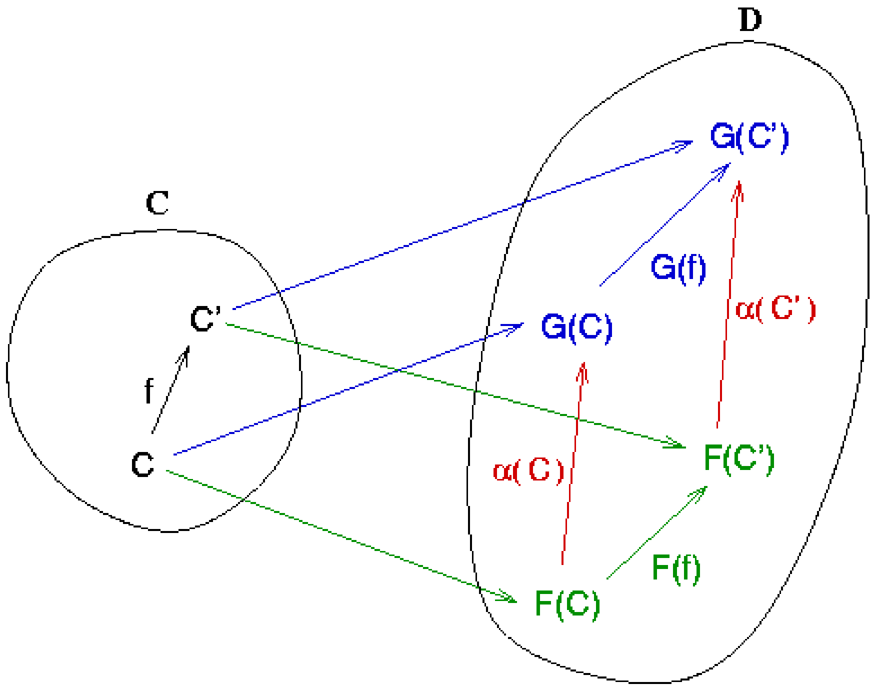
- $\forall\ C \in \mathbf{C}\ \exists\ F(C) \in \mathbf{D}$
- $F(Id_C) = Id_{F(C)}$
- and the diagram

$$\begin{tikzcd} F(C) \arrow[r, "F(f)"] \arrow[dr, "F(g \circ f)"'] & F(C') \arrow[d, "F(g)"] \\ & F(C'') \end{tikzcd}$$

is commutative

Natural transformation

$\alpha : F \longrightarrow G$



Two functors:

$F : \mathbf{C} \longrightarrow \mathbf{D}$

$G : \mathbf{C} \longrightarrow \mathbf{D}$

The morphism

$\alpha : F \longrightarrow G$

is natural if the diagram

$$\begin{tikzcd} F(C) \arrow[r, "F(f)"] \arrow[d, "\alpha(C)"'] & F(C') \arrow[d, "\alpha(C')"] \\ G(C) \arrow[r, "G(f)"] & G(C') \end{tikzcd}$$

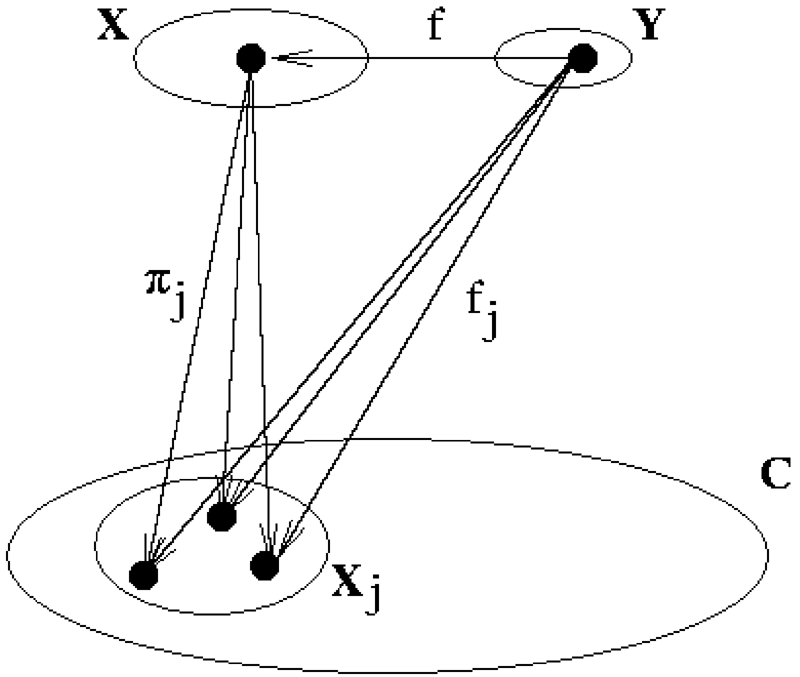
is commutative

Product

an projective cone

$$\mathbf{X} = \prod_{j \in J} \mathbf{X}_j$$

$f_j = f \circ \pi_j$ is unique



a commutative diagram

$$\begin{tikzcd}
\mathbf{X} \arrow[d, "\pi_j"'] & \mathbf{Y} \arrow[l, "f"'] \arrow[dl, "f_j"] \\
\mathbf{X}_j &
\end{tikzcd}$$

a product of morphisms

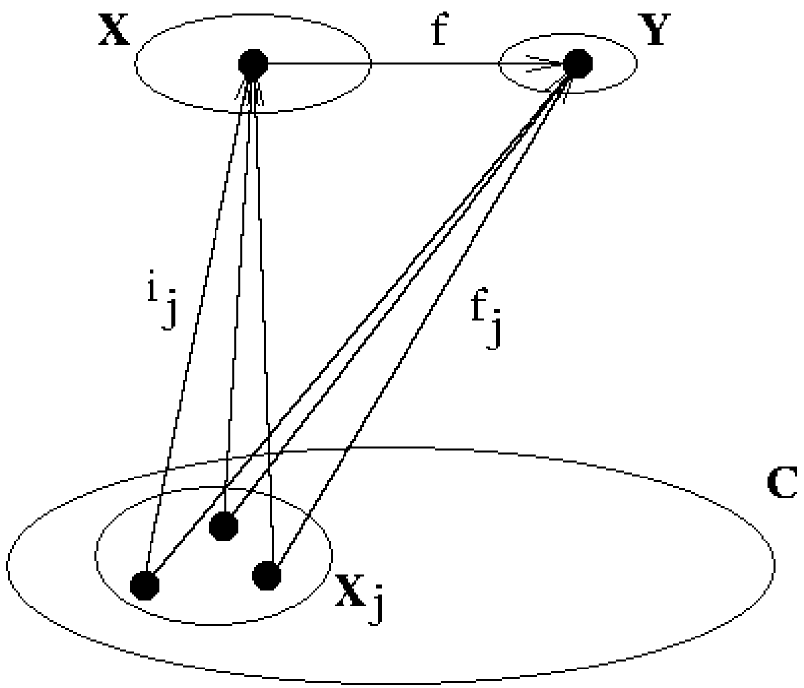
$$\langle f_1 \cdots\cdots f_n \rangle$$

Coproduct

an injective cone

$$\mathbf{X} = \bigoplus_{j \in J} \mathbf{X}_j$$

$f_j = f \circ i_j$ is unique



a commutative diagram

$$\begin{tikzcd} \mathbf{X} \arrow[r, "f"] & \mathbf{Y} \\ \mathbf{X}_j \arrow[u, "i_j"] \arrow[ur, "f_j"'] & \end{tikzcd}$$

Coproduct = colimit

an inductive limit

$$\mathbf{X} = \varinjlim \mathbf{X}_i$$

a direct set $\langle I, \leqslant \rangle$

a direct system $\langle \mathbf{X}_i, f_{ij} \rangle$

a disjoin union $\mathbf{X} = \varinjlim \mathbf{X}_i$

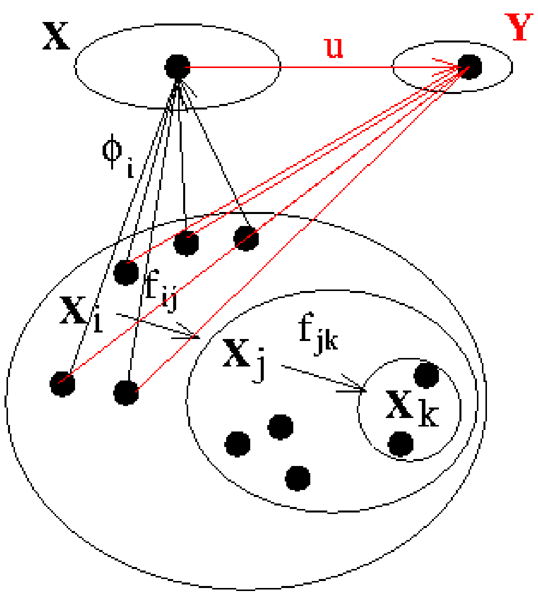


diagram commutes $\forall$ i,j

$$\begin{tikzcd}
\mathbf{X}_i \arrow[rr, "f_{ij}"] \arrow[dr, "\phi_i"] \arrow[ddr, "\psi_i"'] & & \mathbf{X}_j \\
& \mathbf{X} \arrow[d, "u"] & \\
& \mathbf{Y} &
\end{tikzcd}$$

u: $\mathbf{X} \longrightarrow \mathbf{Y}$ is unique $\forall$ i,j

Direct limit = colimit

an inductive limit

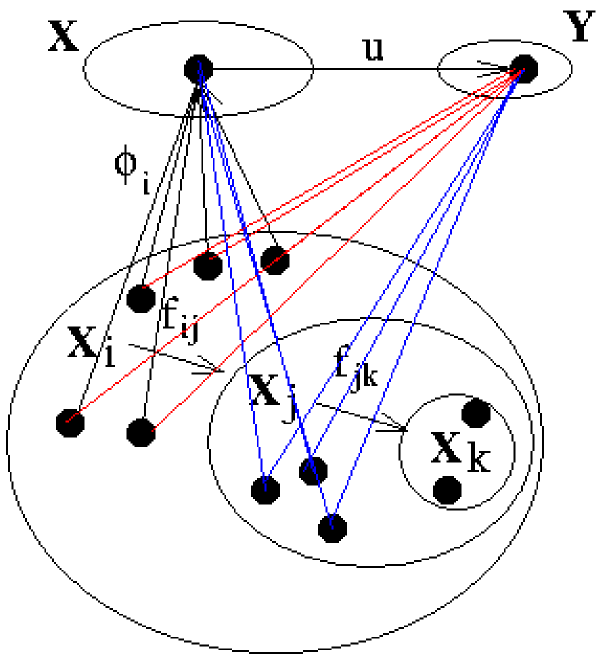
$$\mathbf{X} = \varinjlim \mathbf{X}_i$$

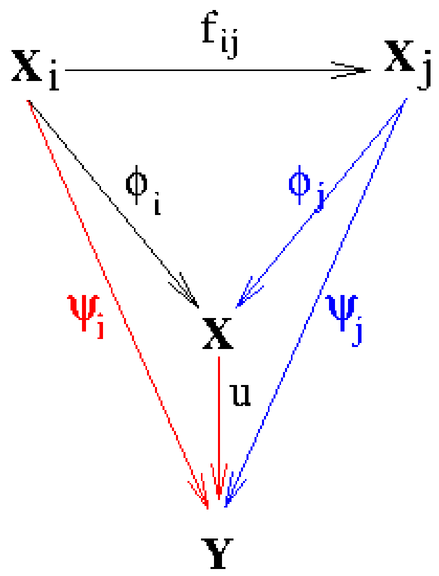
a direct set $\langle I, \leqslant \rangle$

a direct system $\langle \mathbf{X}_i, f_{ij} \rangle$

a disjoin union $\mathbf{X} = \varinjlim \mathbf{X}_i = \bigoplus_i \mathbf{X}_i / \sim$

diagram commutes $\forall$ i,j





u: $\mathbf{X} \to \mathbf{Y}$ is unique $\forall$ i,j

Data base axes

- Relation
  Relational model: Hom$_{Set^{Rel}}$ is empty
  Example: Table<Antenna>

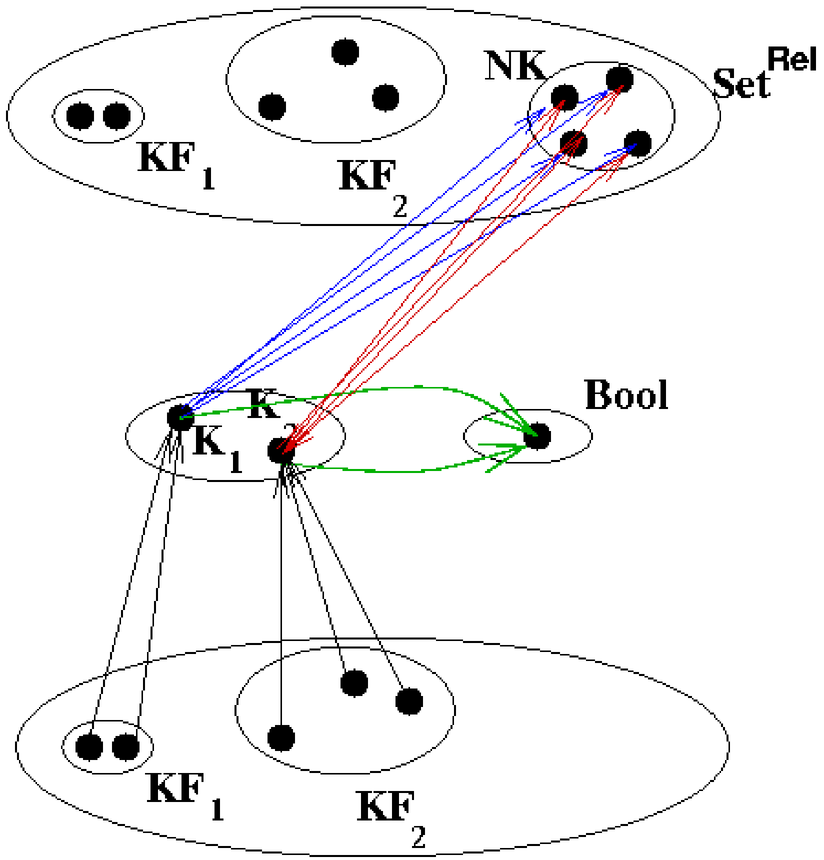
- Recursion
  Use-cases require a cone or co-cone (projection or induction)
  Example: Table<Antenna> (in case of APA and PAF)

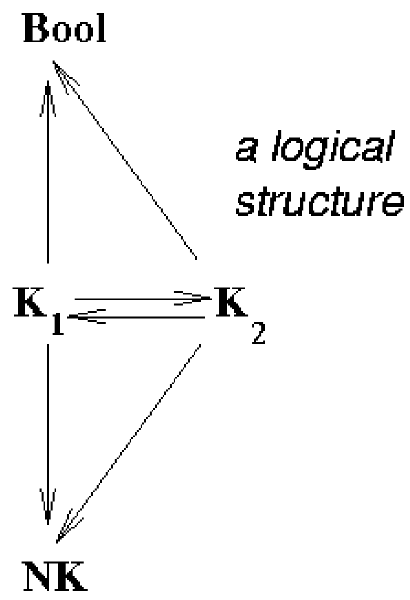
- Monoids: A table is a data base
  $Mon(Set^{Rel}, +)$

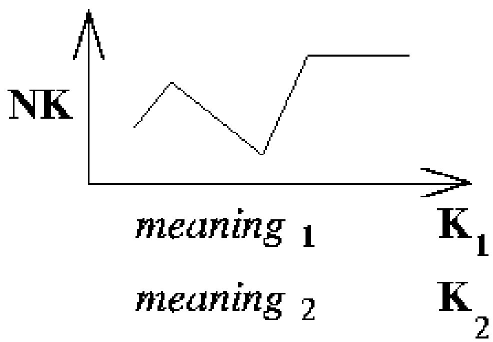
Relation *the category* $\mathbf{Set}^{\mathbf{Rel}}$

Defined using projective and injective cones

Must conform to the normal forms (logic based)



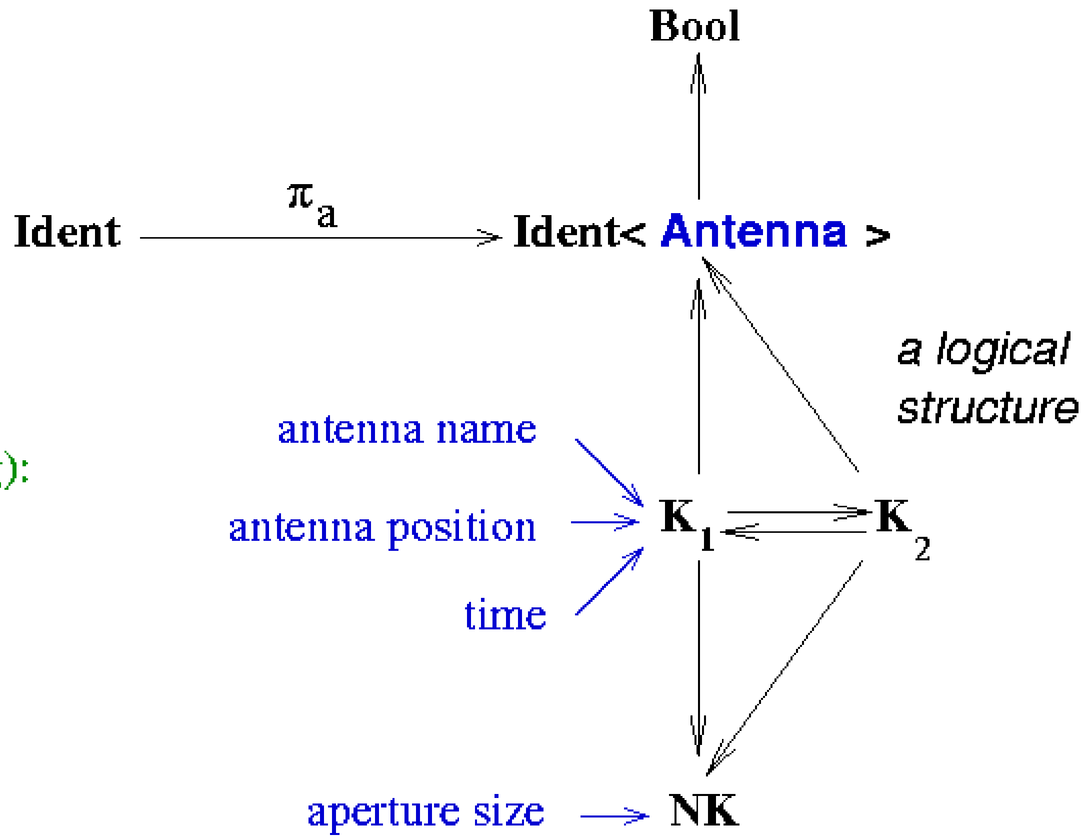




Relation Table< Antenna >

Primary key (meaning):

array geometry



Relation Table< Antenna >

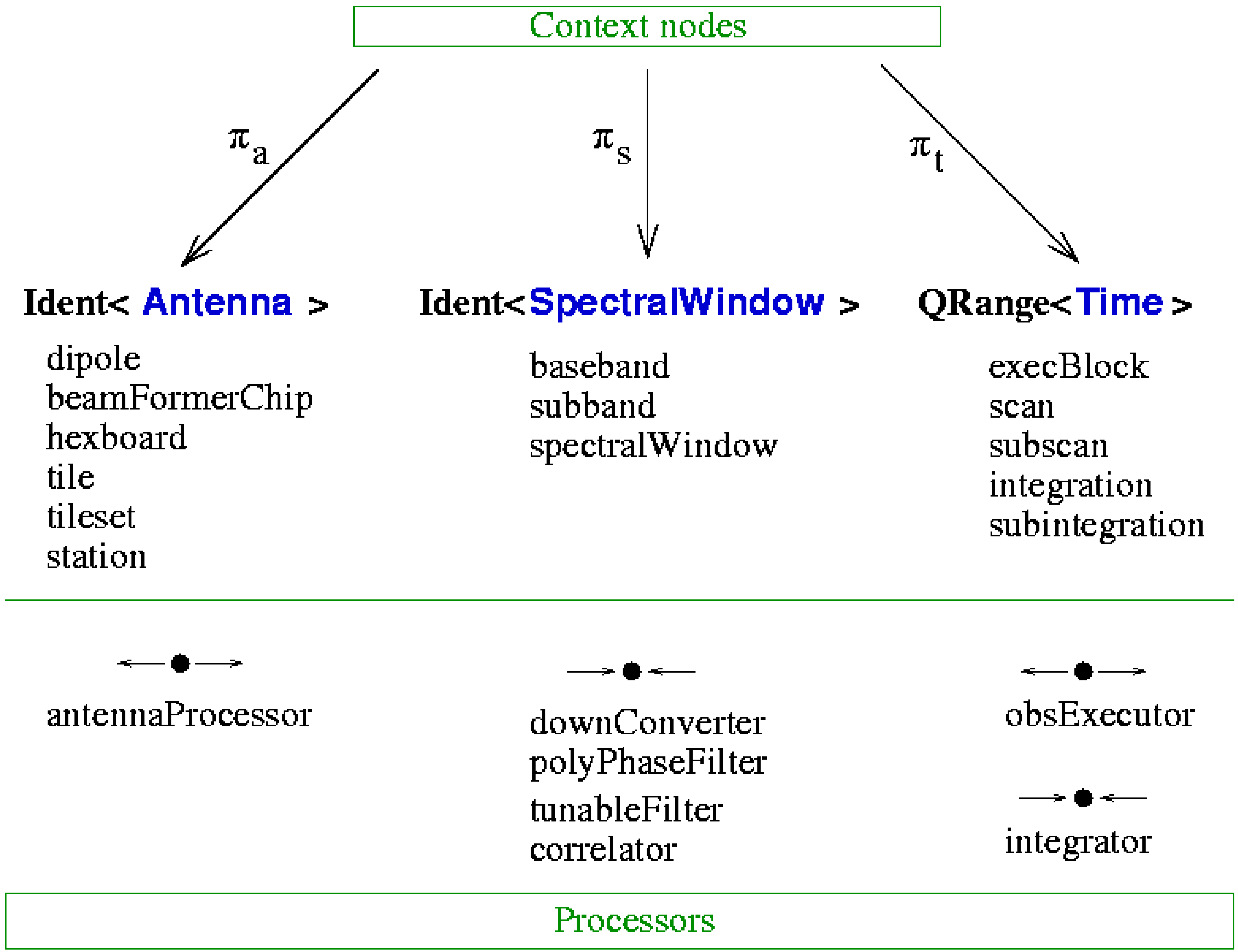
$$
\begin{tikzcd}
& & & \mathbf{Bool} & \\
\mathbf{Ident} \arrow[rrr, "\pi_a"] & & & \mathbf{Ident}\langle \textbf{Antenna} \rangle_{k>j} \arrow[u] & \\
& & \text{antenna name} \arrow[dr] & & \textit{a logical structure} \\
& & \text{antenna position} \arrow[r] & \mathbf{K}_1 \arrow[uu] \arrow[r, shift left] \arrow[dd] & \mathbf{K}_2 \arrow[l, shift left] \arrow[uul] \arrow[ddl] \\
& & \text{time} \arrow[ur] & & \\
& & & \mathbf{NK} & \\
& & \prod_{j \in J} \mathbf{Ident}\langle \textbf{Antenna} \rangle_j \arrow[ur] & & \mathbf{PQ}\langle \mathbf{Length} \rangle \arrow[ul] \\
& & \text{aperture components} & & \text{aperture size}
\end{tikzcd}
$$

Use–case of APA (e.g. EMBRACE)

Antenna $\twoheadrightarrow$ Antenna components

A recursive object, a projective limit

Topological space axis basis



Monad

Let consider a functor $T: C \longrightarrow C$ and 2 natural transformations:

the induction η_t: ident $\longrightarrow$ T(ident)

the direct system μ_t: T(T(ident)) $\longrightarrow$ T(ident)

We have:

the monad (T, η_t, μ_t)

$$\begin{tikzcd} \mathbf{T}^3 \arrow[r, "\mu_t T"] \arrow[d, "\mu_t T"'] & \mathbf{T}^2 \arrow[d, "\mu_t"] \\ \mathbf{T}^2 \arrow[r, "\mu_t"'] & \mathbf{T} \end{tikzcd} \qquad \begin{tikzcd} \mathbf{T} \arrow[r, "T\eta_t"] \arrow[d, "\eta_t T"'] \arrow[dr, equal] & \mathbf{T}^2 \arrow[d, "\mu_t"] \\ \mathbf{T}^2 \arrow[r, "\mu_t"'] & \mathbf{T} \end{tikzcd}$$

its T–algebra(t,h)

$$\begin{tikzcd} \mathbf{T}^2 t \arrow[r, "Th"] \arrow[d, "\mu_t"'] & \mathbf{T} t \arrow[d, "h"] \\ \mathbf{T} t \arrow[r, "h"'] & t \end{tikzcd} \qquad \begin{tikzcd} t \arrow[r, "\eta_t"] \arrow[dr, "1_t"'] & \mathbf{T} t \arrow[d, "h"] \\ & t \end{tikzcd}$$

Application: Table< Feed >

Proposition: A table is a monad which has for its algebraic structure the vector space, the directed set

$$\mathrm{Ident}\langle \mathrm{Antenna} \rangle \longrightarrow \mathrm{Ident}\langle \mathrm{SpectralWindow} \rangle \longrightarrow \mathrm{QRange}\langle \mathrm{Time} \rangle$$

$$\begin{tikzcd}
\mathrm{Ident}\langle \mathrm{Antenna} \rangle & & \\
& \mathrm{Ident}\langle \mathrm{SpectralWindow} \rangle \arrow[ul] & \\
& \mathrm{QRange}\langle \mathrm{Time} \rangle \arrow[u] & \\
\mathrm{Ident}\langle \mathrm{Feed} \rangle \arrow[ur] & & \mathrm{Relation}\langle \mathrm{Feed} \rangle \arrow[ul]
\end{tikzcd}$$

Application: Table< Feed >

Proposition: A table is a monad which has for its algebraic structure the vector space, the directed set

Ident<Antenna> ⟶ Ident<SpectralWindow> ⟶ QRange<Time>

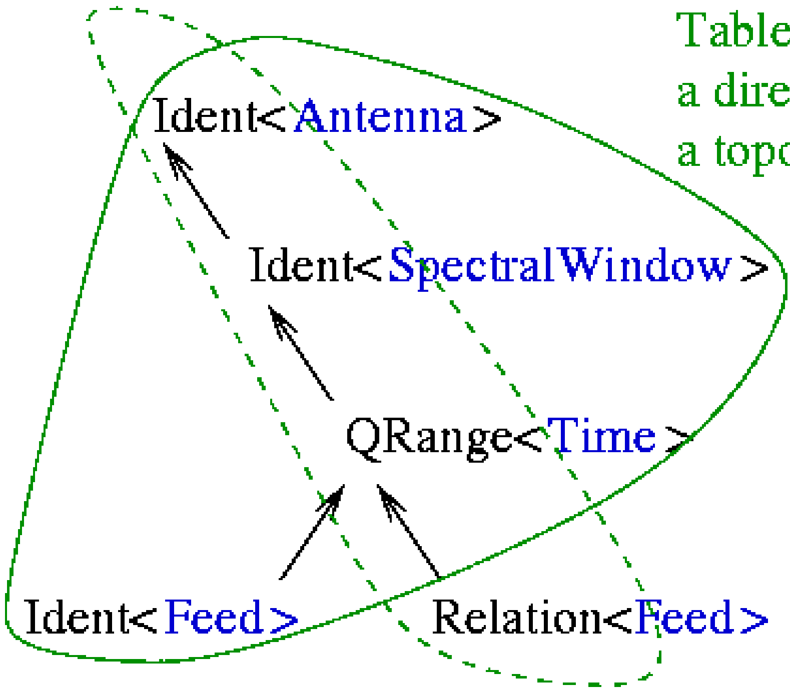


Table key section:
a directed product,
a topological space for the algebra $\uplus$

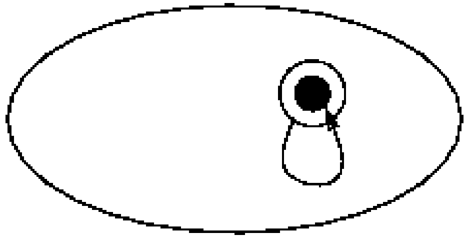
New compared to casa:MS
- *col are ordered in the key section*
- *the data section has a primary key*
- *must be a set*

Like asdm:ASDM
- *a database (uniqueness)*

New compared to asdm:ASDM
- *table with sections like in MS*
- *the data section has a primary key*

A relation in context *(also normalized)*

Tables, singletons, are monads in a category Dataset



Dataset<Sdm>

Category *Profile*

also a monad

Category

Example: Table< Feed > in the SDMv2

$$
\begin{tikzcd}
 & \text{Ident<Antenna>} & \\
 & \text{Ident<SpectralWindow>} \arrow[u, "\pi"] & \\
 & \text{QRange<Time>} \arrow[u, "\pi"] & \\
\text{Ident<Feed>} \arrow[ur] & & \text{Relation<Feed>} \arrow[ul, "\pi"']
\end{tikzcd}
$$

Construction:

a $\lambda-$ calculus

an immersion in a topological space

Where is the meaning?

We have seen that a relation is the composition of attributes.
Relations are named and their meanings are bound to constraints, rules of constructions.

An attribute: association "column name,type" *(NB: relational model: type=PDT)*.

Names are words, a subset of the set **String**.
In practice very convenient but at the cost of several deficiencies:

1. **Not efficient**: parsing string is expensive. Solutions to this are:

   carry the meaning via names of variables and use PDT

   but

2. **Not type-safe**: codes not robust
3. **Meanings embedded in codes**: semantic embedded in the codes
4. **Tendency to use implicit words**: information 'hidden' in documentation

Proposition:Attribute *(in relation)* must rely on data models.

at the cost of departures from the relational data model

Type specification

A type is determined by a domain.
a set which gives all the possibles values:

basis of a type.

SDMv2 data types have data models:

basis + logical structure + algebraic structure

Type specification: QValue(s) data model:

1. Abstract QValue data model

   - A value is parametrized by a primitive data type
   - A value may be *undefined* or *actual* or *virtual*
     virtual values are expressions parametrized by a function type:

     POLYNOMIAL, STAIRCASE, CHEBYSHEV

2. Abstract QValues data model

   - A collection of values has meta-data to describe its structure
     Any collection of values is parametrized by a dimensionality to discriminate **scalars** from **vector**s
   - The actual content of a collection is independent of its representation.
     there is a **polymorphic representation to support data compression**.
   - Collections have iterators

Type specification: Simple use-case: PQuantity

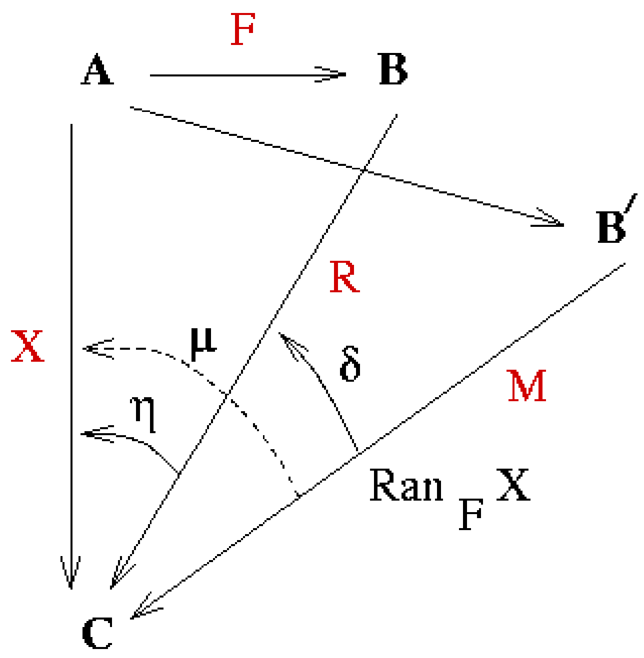
1. A categorical construct: the Kan extension

2. Handling multiple system of units

3. Its topology: a 2-category on a vector space

4. Its algebraic structure to get its categorical theory

Kan extension



3 categories, **A**, **B**, **C**

3 functors, F, M, X

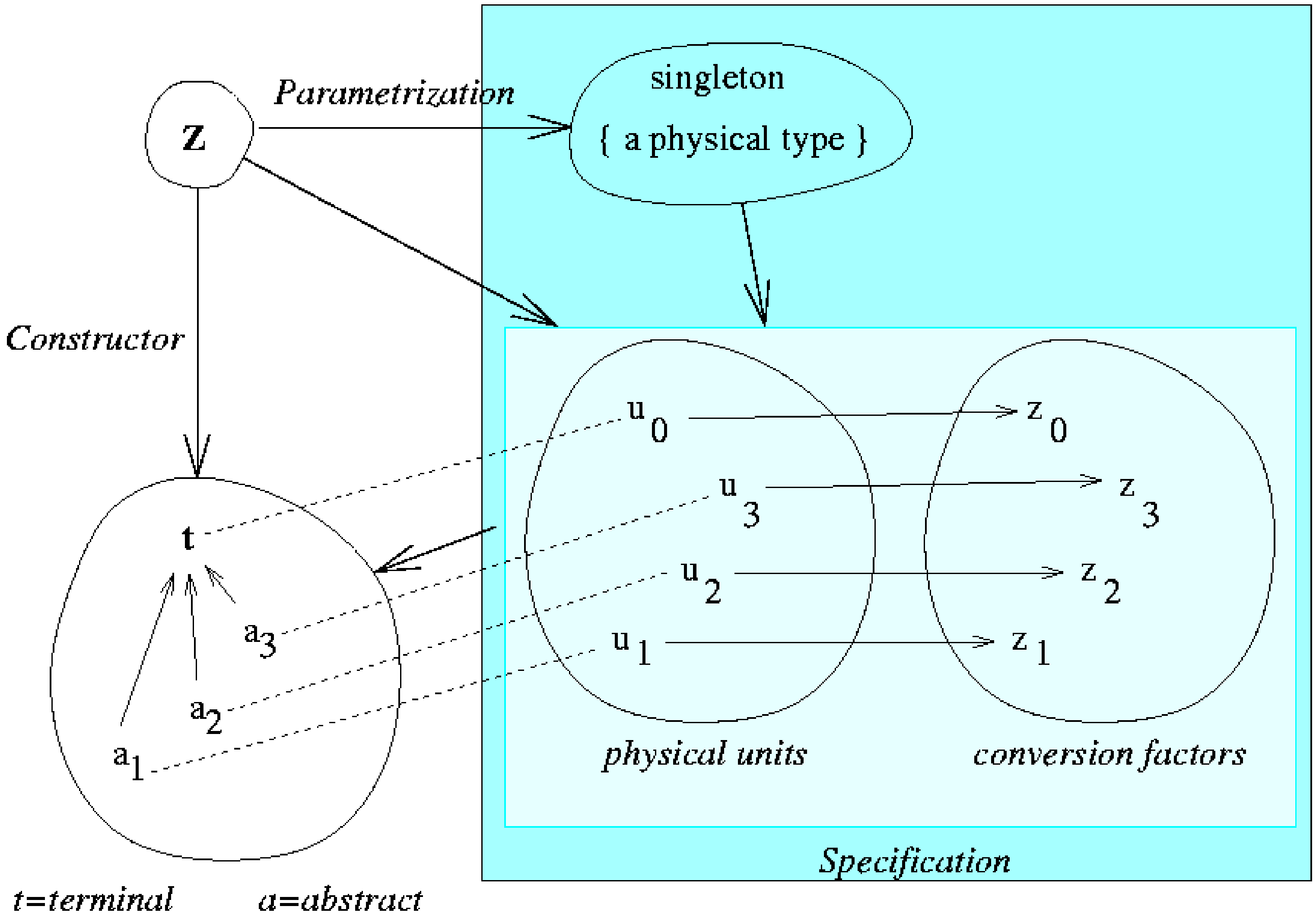
2 natural transformations η, δ

$\forall$ M : **B** $\longrightarrow$ **C**

Let μ be a natural transformation

$\forall$ μ : M F $\longrightarrow$ X

==> δ : M $\longrightarrow$ R is unique



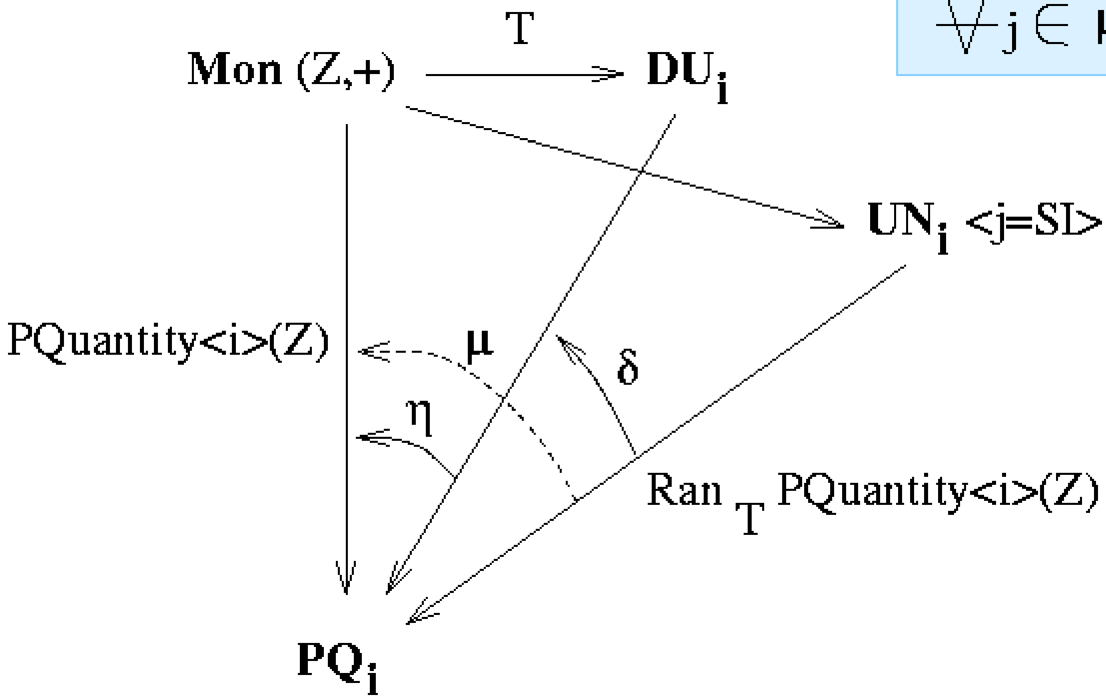
Parametrization
z
singleton
{ a physical type }
Constructor
t
a_3
a_2
a_1
u_0
u_3
u_2
u_1
z_0
z_3
z_2
z_1
physical units
conversion factors
Specification
t=terminal
a=abstract

PQuantity data type

$\delta_i : \text{units}_i \longrightarrow \text{canonic units}_i$

$\forall\, i \in \{\ \text{PQname}\ \}$

$\forall\, j \in \mu_{i,j} : \text{units}_i\,\{j\} \longrightarrow Z$



PQuantity

Let $\mathbf{U}_j$ a unit element along the axis j in a vector space $\mathbf{B}$ an object $\in$ **Vect**

Let $\mathbf{NC}_j \in \mathbf{R}$ the dimension unit along the axis j of a point in that space

$$
\begin{tikzcd}
\mathbf{B} \arrow[dr, "\pi_j"] & & \\
 & \mathbf{U}_j & \\
 & \mathbf{NC}_j \arrow[u, "u_j"] \arrow[r] & \mathbf{PQ}_j \arrow[ul, "u_j"']
\end{tikzcd}
$$

Examples with j=0

Phys quan. k	$\mathbf{NC}_{0,k}$
Length	*1*
Area	*2*
SpatialFrequency	*−1*

PQuantity

Let $\mathbf{U}_j$ a unit element along the axis j in a vector space $\mathbf{B}$ an object $\in$ **Vect**

Let $\mathbf{NC}_j \in \mathbf{R}$ the dimension unit along the axis j of a point in that space

Let $\mathbf{SC}_j \in \mathbf{R}$ the dimension unit ratio along the axis j of a point in that space

$$\begin{tikzcd}
\mathbf{B} \arrow[dr, "\pi_j"] & & \\
& \mathbf{U}_j & \mathbf{SC}_j \arrow[l, "u_j"'] \arrow[d] \\
& \mathbf{NC}_j \arrow[u, "u_j"] \arrow[r] & \mathbf{PQ}_j \arrow[ul, "u_j"']
\end{tikzcd}$$

Examples with j=0

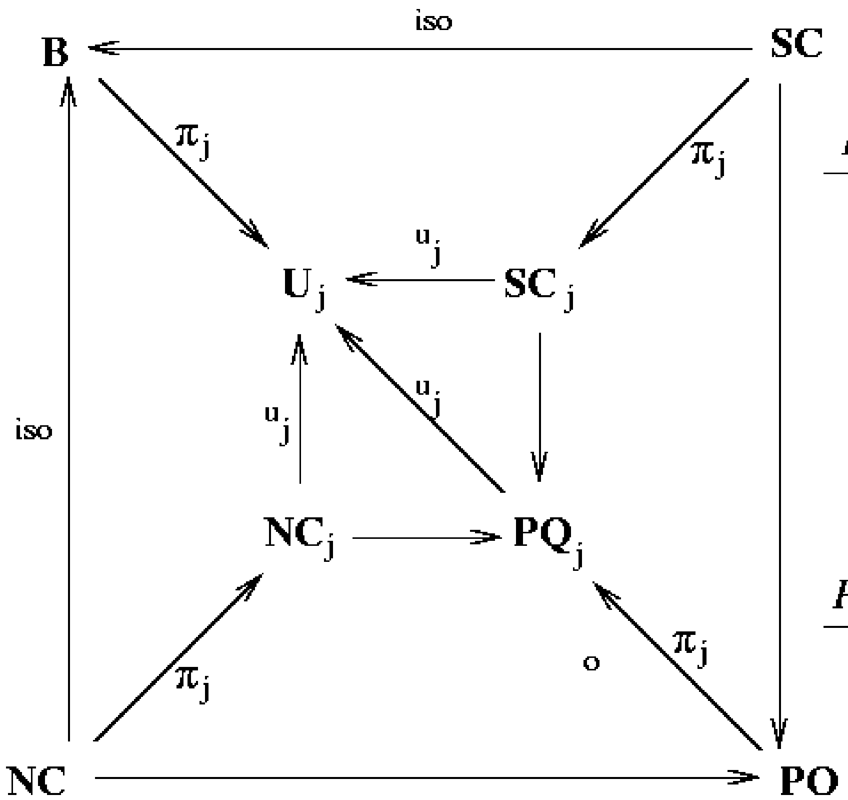
Phys quan. k	$\mathbf{NC}_{0,k}$	$\mathbf{SC}_{0,k}$
Length	*1*	*0*
Area	2	*0*
SpatialFrequency	-1	*0*
Angle	*0*	*1*
SolidAngle	*0*	2
RotationMeasure	-2	*1*

PQuantity

Let $\mathbf{U}_j$ a unit element along the axis j in a vector space $\mathbf{B}$ an object $\in$ **Vect**

Let $\mathbf{NC}_j \in \mathbf{R}$ the dimension unit along the axis j of a point in that space

Let $\mathbf{SC}_j \in \mathbf{R}$ the dimension unit ratio along the axis j of a point in that space



Examples with j=0

Phys quan. k	$\mathbf{NC}_{0,k}$	$\mathbf{SC}_{0,k}$
Length	*1*	*0*
Area	*2*	*0*
SpatialFrequency	*−1*	*0*
Angle	*0*	*1*
SolidAngle	*0*	*2*
RotationMeasure	*−2*	*1*

Examples with j=0 and 2

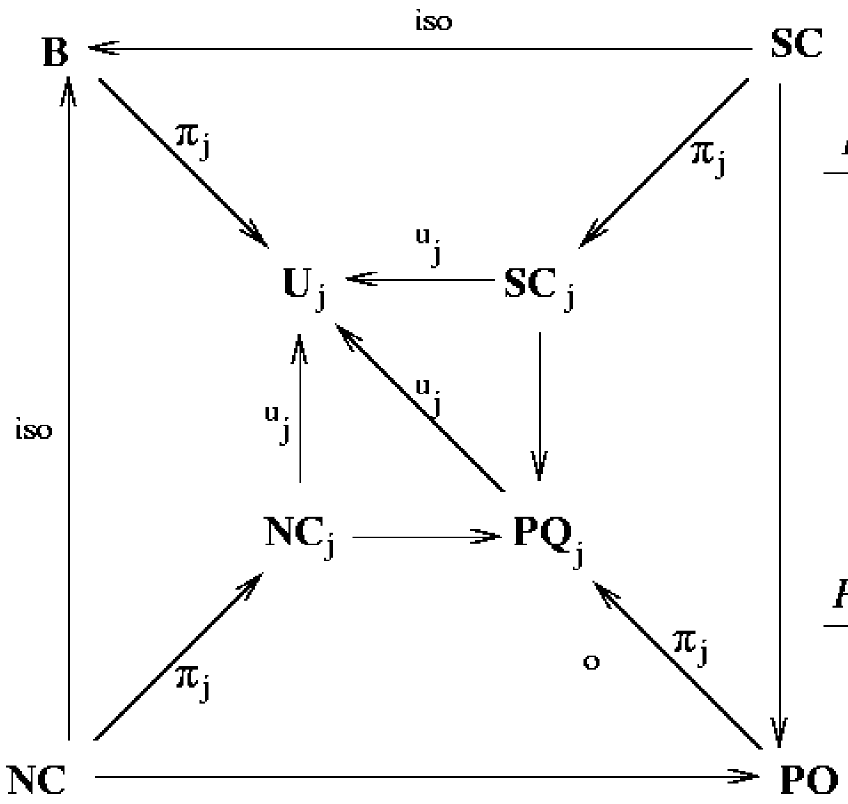
Phys quan. k	$\mathbf{NC}_{0,k}$	$\mathbf{SC}_{0,k}$	$\mathbf{NC}_{2,k}$
Speed	*1*	*0*	*−1*
AngularRate	*0*	*1*	*−1*

PQuantity

Let $\mathbf{U}_j$ a unit element along the axis j in a vector space $\mathbf{B}$ an object $\in$ **Vect**

Let $\mathbf{NC}_j \in \mathbf{Q}$ the dimension unit along the axis j of a point in that space

Let $\mathbf{SC}_j \in \mathbf{Q}$ the dimension unit ratio along the axis j of a point in that space



Examples with j=0

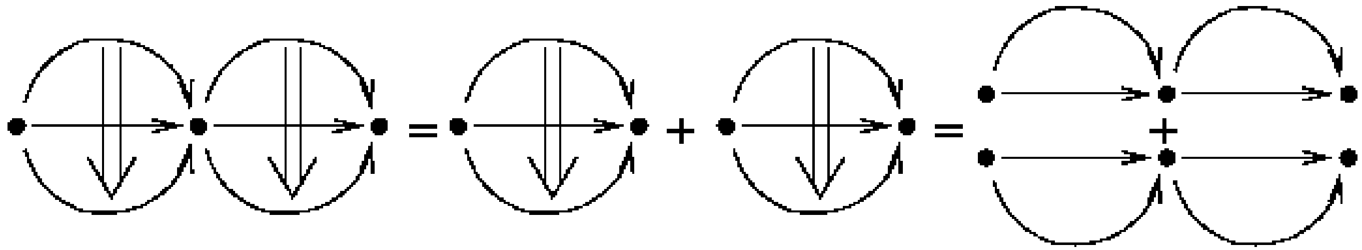
Phys quan. k	$\mathbf{NC}_{0,k}$	$\mathbf{SC}_{0,k}$
Length	1	0
Area	2	0
SpatialFrequency	−1	0
Angle	0	1
SolidAngle	0	2
RotationMeasure	−2	1

Examples with j=0 and 2

Phys quan. k	$\mathbf{NC}_{0,k}$	$\mathbf{SC}_{0,k}$	$\mathbf{NC}_{2,k}$
Speed	1	0	−1
AngularRate	0	1	−1
NEP			−1/2

PQuantity

The topology of PQuantity is a 2–category on a vector space.



this vector space is of dimension 7:

0 Length
1 Mass
2 Time
3 Temperature
4 LuminousIntensity
5 MolarConcentration
6 ElectricCurrent

the horizontal composition along the foundamental physical unit basis
the vertical composition for the dimension,dimensionless property

PQuantity is a monoid for the addition.

Which diagram we need to add to get the theory?

Physical expressions require the addition and the multiplication (ring)

Let consider the following composite graph:
2 monoids PQ_a and PQ_c and the functor m such that $PQ_c = PQ_a \times PQ_b$

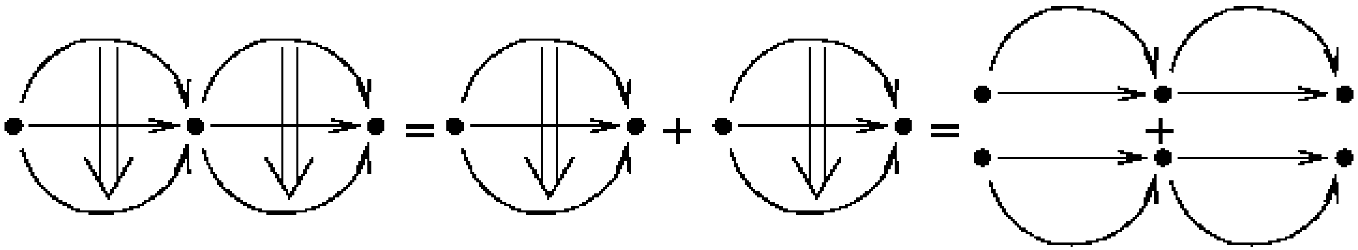
This functor requires to define an other functor for the DU algebra: $DU_c = DQ_a \times DU_b$, a product in the topologic space.

Proposition: This functor is the product of 7 4-simplices (pyramides)

NB: this requires using the ternary operator “?:” to be constructible.

PQuantity

The topology of PQuantity is a 2–category on a vector space.

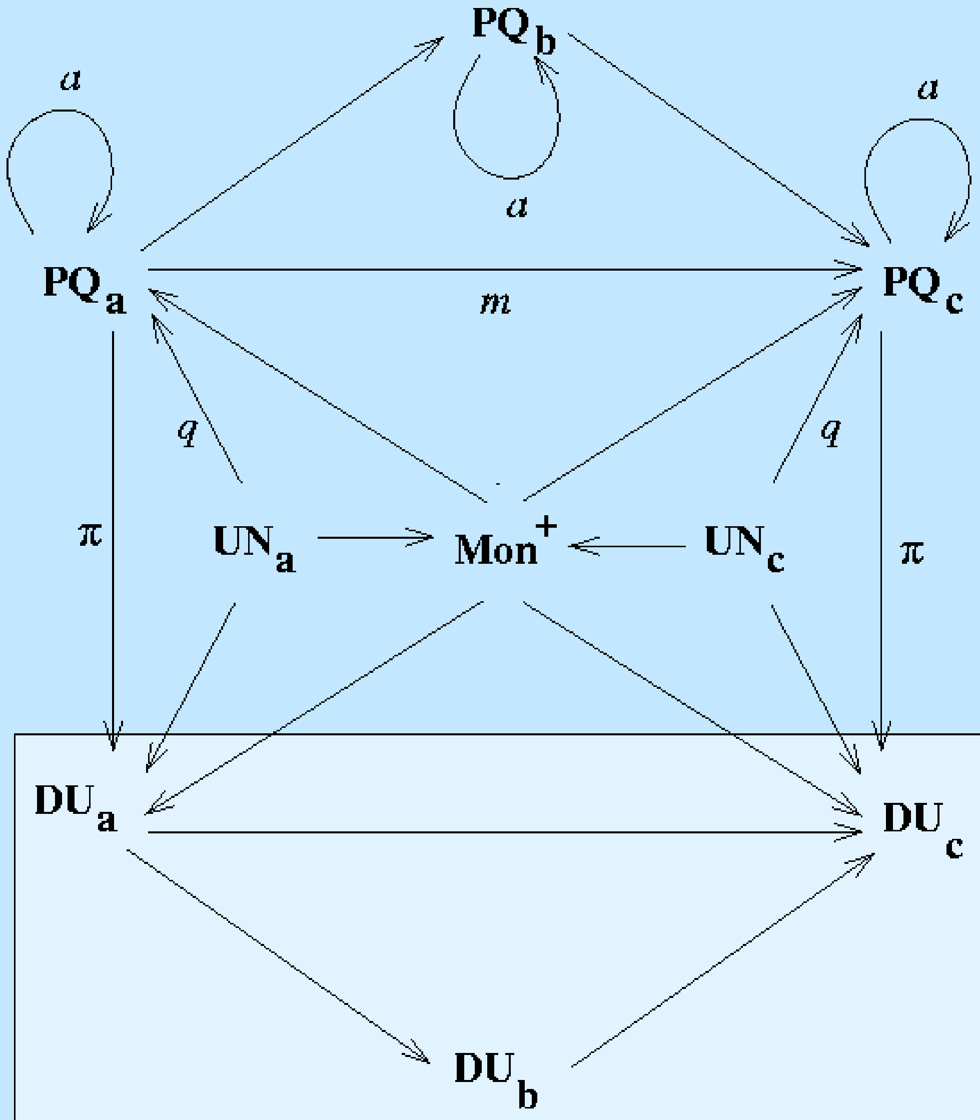


this vector space is of dimension 7:

0 Length
1 Mass
2 Time
3 Temperature
4 LuminousIntensity
5 MolarConcentration
6 ElectricCurrent

the horizontal composition along the foundamental physical unit basis
the vertical composition for the dimension,dimensionless property

PQuantity is a monoid for the addition.



PQ

a type–safe monad
(endomorphism)
(functional language)

m is a projective cone

Sketch of PQuantity

topology:

algebraic data types
calculus at compile time

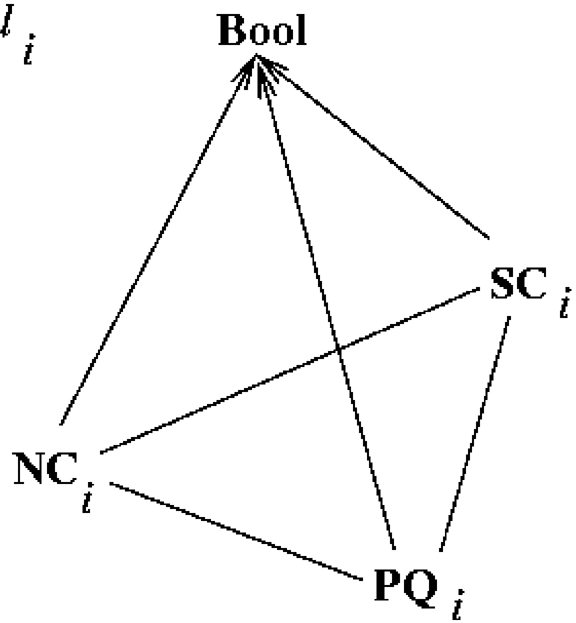
concept of dimension
==> *projective cones*

concept of dimensionless
==> *inductive cones*

The Multiplier functor of PQuantity

An algebraic type with a closure: $SC_i / SC_i = Id_i$

Identity element: $Id = \bigoplus_{j \in J} Id_j$



A 3–simplex:

Space	Regions in the DSL
2D facette NC,PQ,Bool	sub–category of the dimensionned PQ
2D facette SC,PQ,Bool	sub–category of the dimensionless PQ
3D volume	category PQ: general case

Conclusions

1. The theory of the measurement set has been mostly developed
2. It is a data model 'above' the relational model
3. It encapsulates the relational model
4. Tables are sets containing a subset of their powersets, allow recursive definitions
5. Tables are topos
6. Tables are monads
7. The measurements are the object of a category
   (allows functional programming)
8. The DM provides a rigorous DSL allowing expressive codes
9. The DM implies very robust codes (minimum of exceptions, type-safety i.e. validation at compile time).

10. The DM implies efficient code

11. The theory of categories provides a useful language for clean codes and should help when optimizing for parallel processing

12. Generic programming in C++ allows to express the mathematical formalism

Prospects

1. Consider implementing "Concepts" although not in C++11?

2. Deliver an implementation by the end of 2011 (ALMA CIPT)

3. Fully test the implementation with EMBRACE, edit profiles for other instruments

4. Produce a DM for the image domain using this approach

5. Think about an algebraic tree to work directly with these DMs

6. Generate queries in the context of DBMSs (test of the logic part)