The Contravariancy of Anticipatory Systems

Dimitrios Sisiaridis, Michael Heather & Nick Rossiter
Northumbria University, Newcastle NE2 1XE, UK

Symposium 10: 8th BCSCMsG International Symposium
on Computational Self-Organised
Emergence

CASYS'11 - International Conference on COMPUTING ANTICIPATORY SYSTEMS
HEC Management School
University of Liege, LIEGE, Belgium, August 8-13, 2011

Nature of Anticipation

- Where does anticipation come from?

- By definition strong anticipation resides in the anticipatory system itself.

- If this is so, anticipation is of the nature of the system and forms part of the Universe.

- Anticipation must therefore reside in nature and arise from relationships in nature.

- Likewise from no more than its definition, the Universe consists of entities related one to the other.

- Thus each entity affects every other.

Existence and Cartesian Closure

- Existence is not therefore just a first order effect but needs an inherent higher order formalism to represent such multi-body interdependence.
- The relationship between any pair of entities depends on every possible path between them.
- In category theory this is the property of cartesian closure found in the highest structure possible -- the identity natural transformation designated as the '*topos*'.
- However if every entity is related to every other it follows that the relationship is both ways but not just a simple inverse relationship as appears from the laws of physics.

Duality

- A category **C** of objects and arrows between the objects will have a dual $\mathbf{C}^{op}$ with arrows reversed.
- The whole universal structure of both-ways relationships will then be represented by the product $\mathbf{C}^{op} \times \mathbf{C}$.
- This gives rise to the principle of duality throughout the Universe.

Ubiquity of Duality

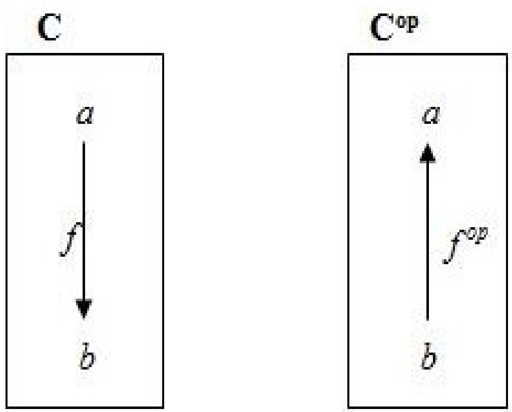
- Duality is a common enough concept in mathematics, philosophy and most of the sciences with some renowned examples like the mind-body duality.
  - It also appears in other versions of contrast as between the dynamic and the static and between global and local.
- To capture the full effect and subtleties of opposing views and relationships a single view of the duality is needed as a process.

Duality and Variance

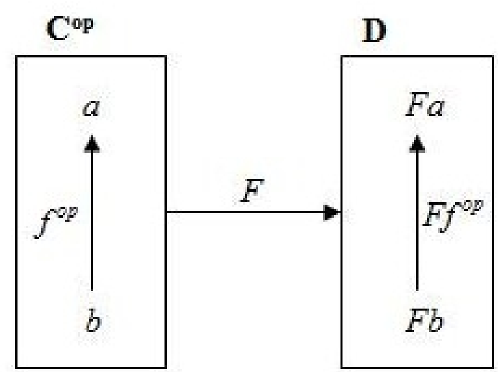
- Duality is not a closed Boolean view.
  - Rather it encapsulates opposite orderings within a single (functorial) concept of variancy.
  - These may be conveniently labelled covariant and contravariant but only relative one to the other and not as absolute descriptions.
- Systems theory is a case in point where these different views need to be integrated.
  - Thus for anticipatory systems, anticipation is an instantaneous, local static instantiation of a dynamic global feature that looks either forward or back.
- The natural categories of process as advanced by Whitehead encompass this contravariancy found in reality.

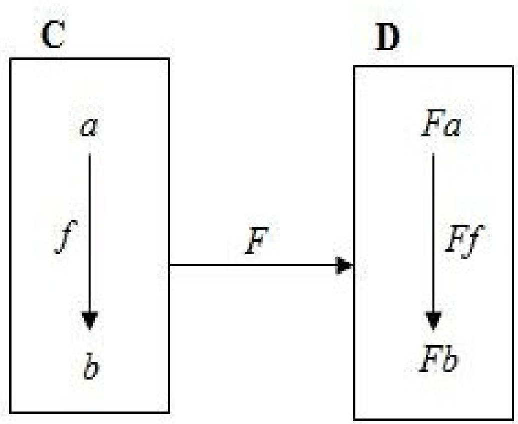
Covariant and Contravariant Functors

Opposite

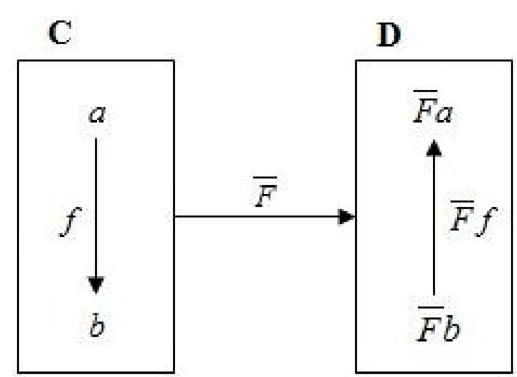


Covariant F





Covariant F



Contravariant F-bar ,

Contravariancy

- Highlighted by Lawvere in 1969 as basic property in the intension-extension relationship
  - Governing data values in the context of their name and type
  - Basic property of universe
- Lawvere defined the relationship between intension and extension in terms of adjoint functors with contravariant mapping
  - Used concept of hyperdoctrine
  - Some 'translation' needed for applied scene

Why Contravariant?

The extension is of the form:

value → *name* N:1

The intension is of the form:

name → *type* N:1

If these arrows were reversed, they would not be functions so can reject such forms:

name → *value* 1:N

type → *name* 1:N

Turn around one arrow

But the common attribute in extension and intension – *name* – is codomain in extension and domain in intension.

So cannot do simple covariant mapping of one to the other.

Need to turn around the arrow in the intension

$$\mathit{name} \rightarrow \mathit{type} \quad \Longrightarrow \quad \mathit{type} \rightarrow \mathit{name}$$

And map this onto *value* → *name* in extension

So that *value* is related to *type* in the context of a common name

Ultimate Contravariancy

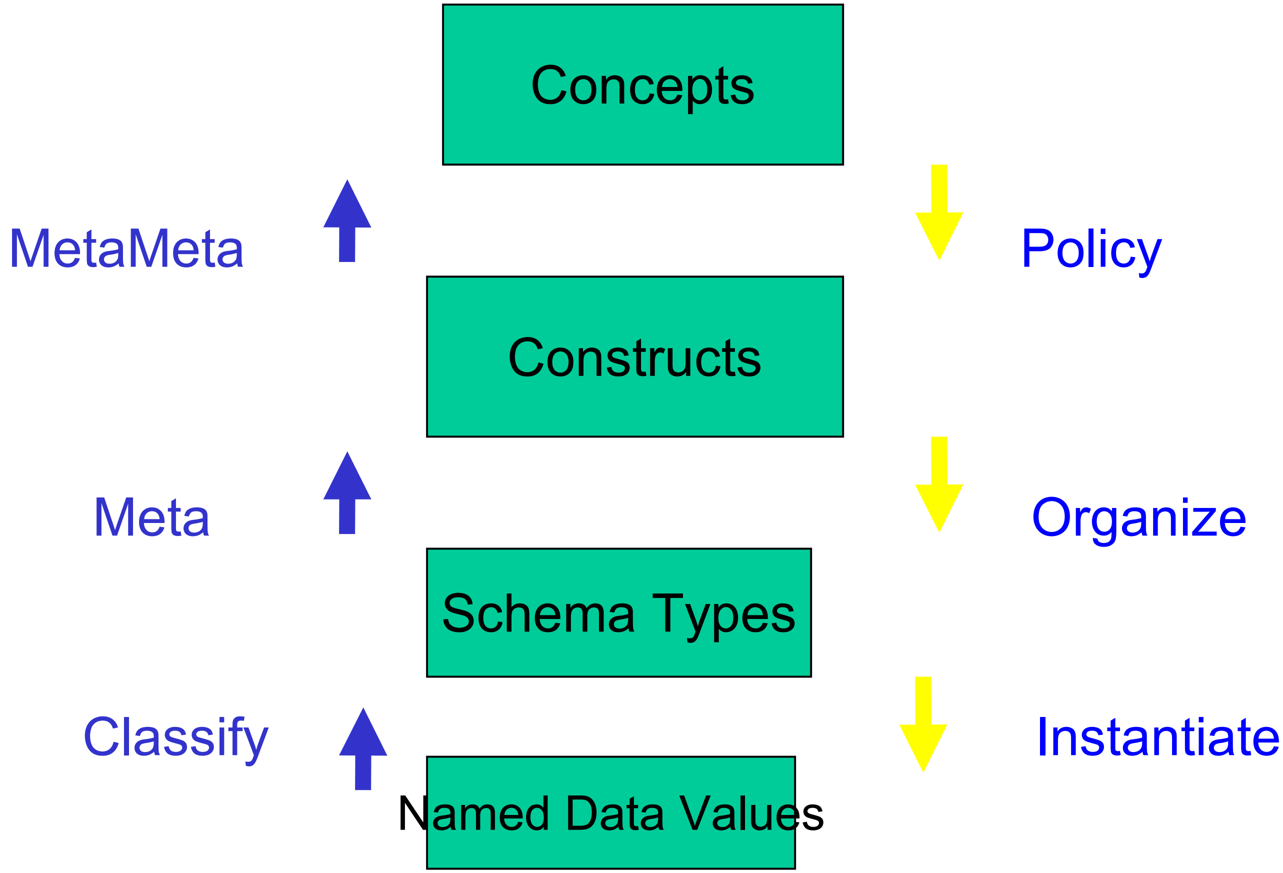
- A three-level structure is sufficient to provide complete closure with internal contravariant logic providing a generalisation of negation.
  – Further levels are redundant
- Contravariancy across levels provides more sophisticated reversals such as reverse engineering.
- The ultimate contravariancy is to be found in the universal adjointness

  - between any pair of functors contravariant one to the other

  - to provide both the quantitative and qualitative semantics of intension-extension logic.

Worked Example of Three-level Architecture

- Choices for realisation in formal terms
- Informal look at structures and relationships
- Outline in informal categories
- Two-way mappings as adjunctions
- Examples of Contravariancy



Downward arrows are intension-extension pairs

Figure 1: Informal requirements for Information System Architecture

Formalising the Architecture

- Requirements:
  - mappings within levels and across levels
  - bidirectional mappings
  - closure at top level
  - open-ended logic
  - relationships (product and coproduct)
- Choice: Category theory as used in mathematics as a workspace for relating different constructions

blue – category, red - functor, green - natural transformation

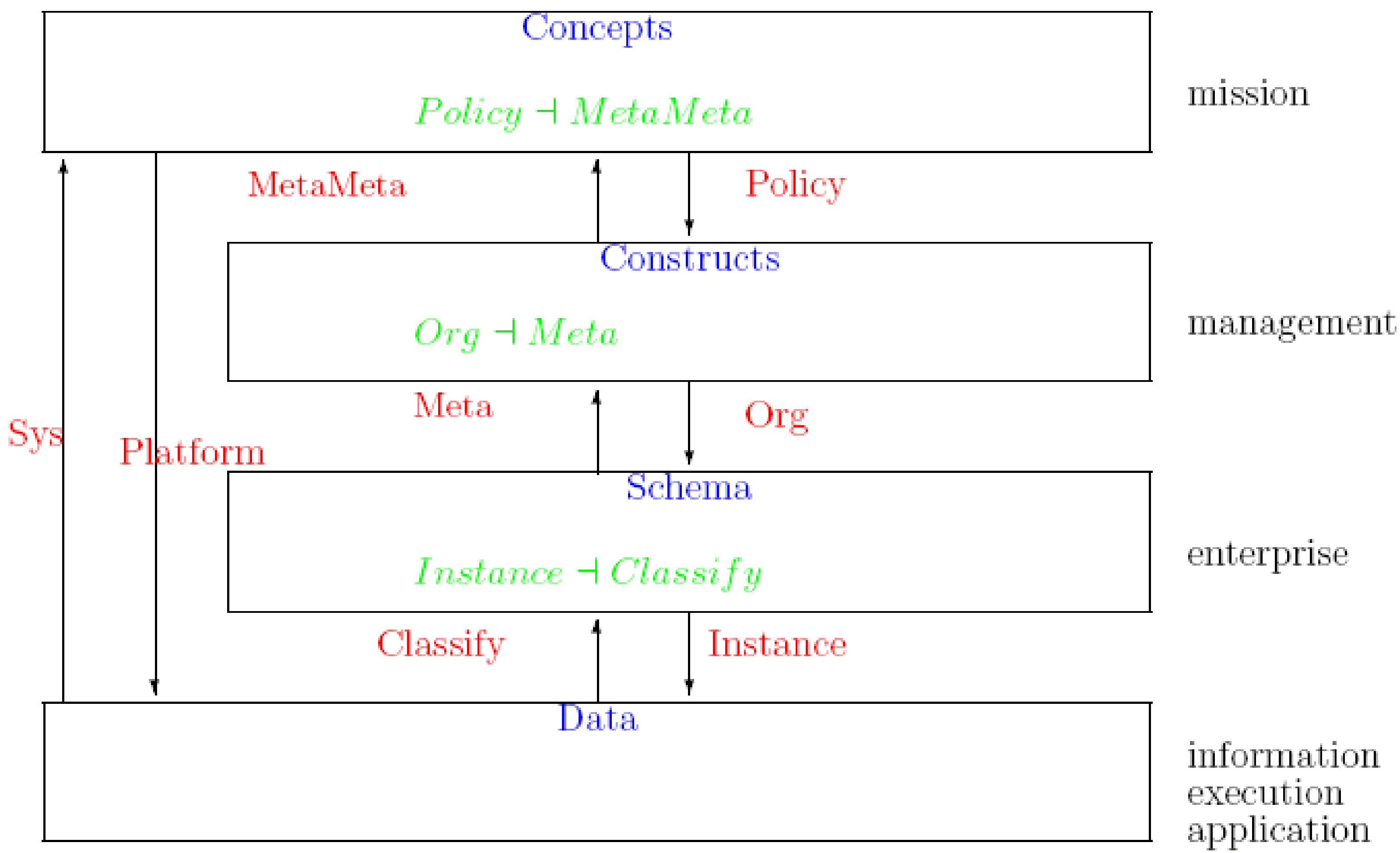


Figure 2: Interpretation of Levels as Natural Schema in General Terms

(Organisational interoperability)

$$
\begin{tikzcd}
 & \mathbf{CST} \arrow[r, "O"'] \arrow[ddr, phantom, "\Downarrow \beta"] & \mathbf{SCH} \arrow[r, "I"'] \arrow[ddr, phantom, "\Downarrow \gamma"] & \mathbf{DAT} \\
\mathbf{CPT} \arrow[ur, "P"] \arrow[dr, "P'"'] \arrow[r, phantom, "\Downarrow \alpha"] & {} & & \\
 & \mathbf{CST}' \arrow[r, "O'"] & \mathbf{SCH}' \arrow[r, "I'"] & \mathbf{DAT}'
\end{tikzcd}
$$

Figure 3: Example for Comparison of Mappings in two Systems
Categories: CPT concepts, CST constructs, SCH schema, DAT data,
Functors: P policy, O org, I instance,
Natural transformations: α, β, γ

black - objects

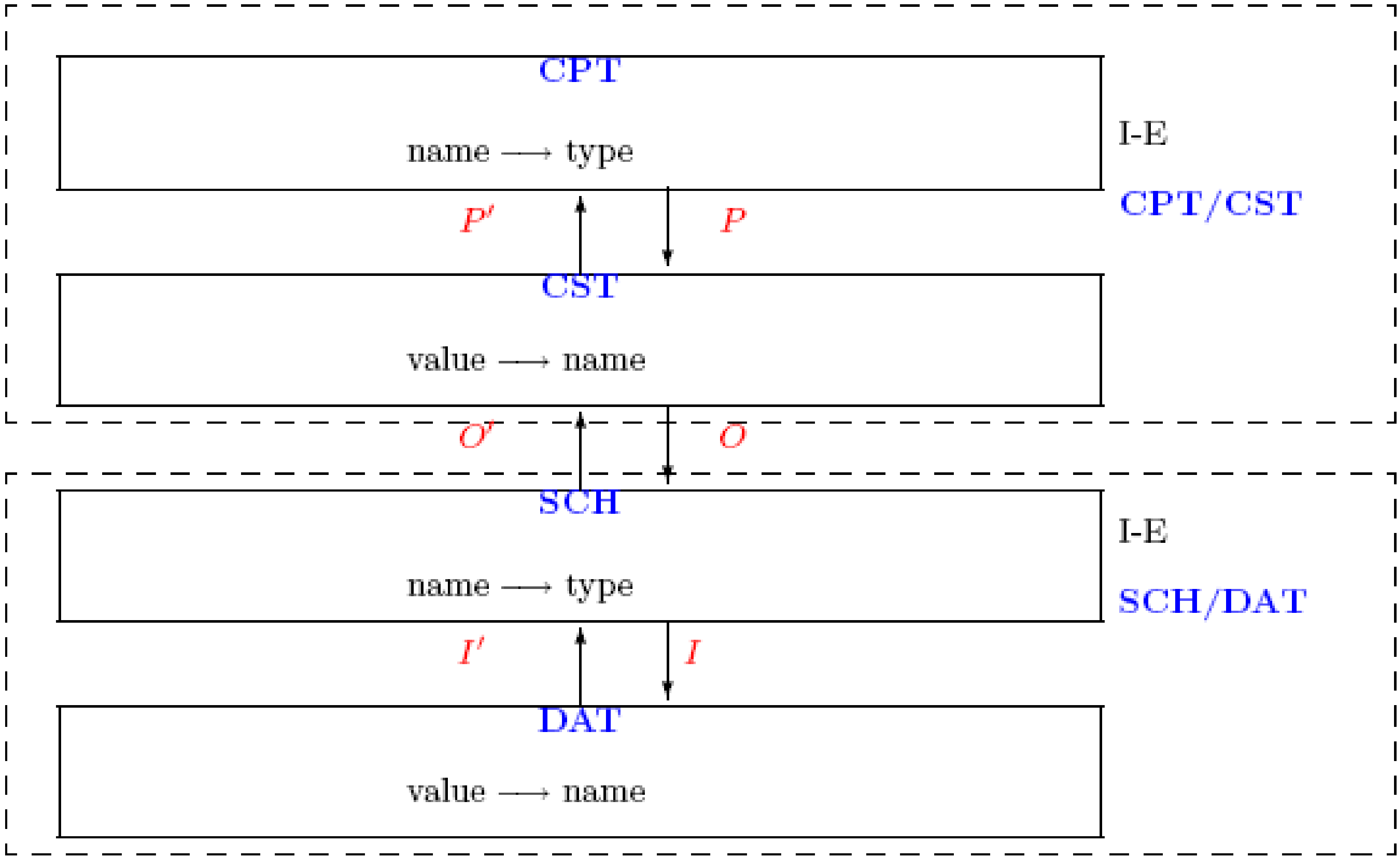


Figure 4: Defining the Three Levels with Contravariant Functors and Intension-Extension (I-E) Pairs

Level	Template	Property	Relational Database (aggregation)	Abstract Data Type (encapsulation)
CPT	name $\longrightarrow$ type	attribute $\longrightarrow$ property	table $\longrightarrow$ aggregation	ADT $\longrightarrow$ encapsulation
P'	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$
CST	value $\longrightarrow$ name	registration_no $\longrightarrow$ attribute	birth_type $\longrightarrow$ table	BST $\longrightarrow$ ADT
O'	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$
SCH	name $\longrightarrow$ type	car_reg $\longrightarrow$ registration_no	birth_record $\longrightarrow$ birth_type	aTree $\longrightarrow$ BST
I'	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$	$\nearrow \quad \nwarrow$
DAT	value $\longrightarrow$ name	'x123yng' $\longrightarrow$ car_reg	<'Smith', 25 mar 1980, 'Torquay' > $\longrightarrow$ birth_record	instance of tree (nodes/links) $\longrightarrow$ aTree

Figure 5: Examples of Levels in the Three Level Architecture

Cross-over arrows indicate contravariant mapping

If functors are adjoint, there is a unique relationship between them (a natural bijection).

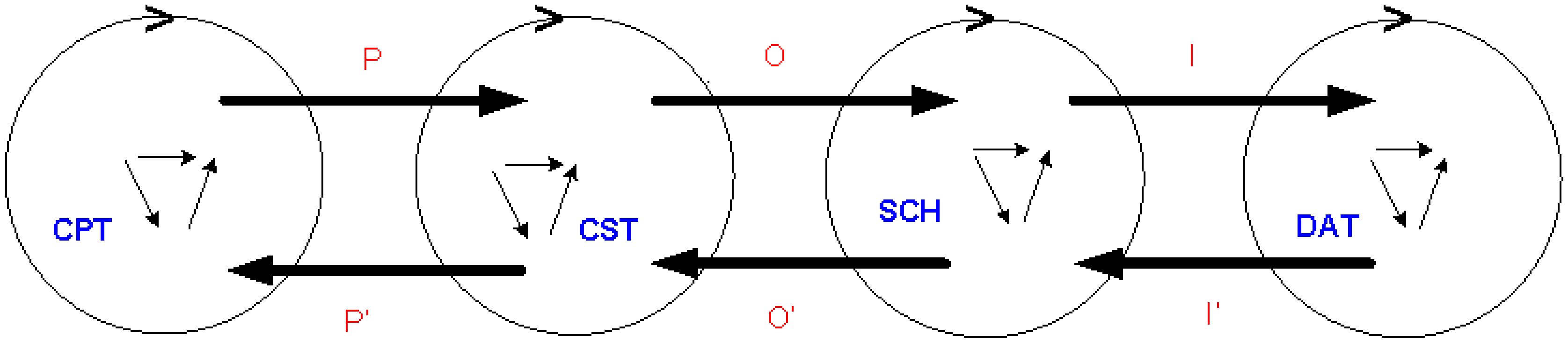


Figure 6: Composition of Adjoints is Natural

Further Work needed on Three-level Architecture

- Lack of detail on
  - Adjoint relationships
  - Underlying categories
- How are they defined?

Unit and Counit of Adjunction between levels DAT and SCH

$$
\begin{tikzcd}
value \arrow[r, "\eta_{value}"] \arrow[dr, "f"'] & GF(value) \arrow[d, "Gg"] & \xrightarrow{F} & F(value) \arrow[d, "Ff"'] \arrow[dr, "g"] & \\
& G(type) & \xleftarrow[G]{} & FG(type) \arrow[r, "\varepsilon_{type}"'] & type
\end{tikzcd}
$$

F --| G

η_{value}: *value* → GF(*value*)

ε_{type} : FG(*type*) → *type*

$$
\begin{tikzcd}
value \arrow[r, "\eta_{value}"] \arrow[dr, "f"'] & G(name) \arrow[d, "Gg"] & \xrightarrow{F} & name \arrow[d, "Ff"'] \arrow[dr, "g"] & \\
& name & \xleftarrow[G]{} & F(name) \arrow[r, "\varepsilon_{type}"'] & type
\end{tikzcd}
$$

F --| G

η_{value}: *value* → G(*name*)

ε_{type} : FG(*name*) → *type*

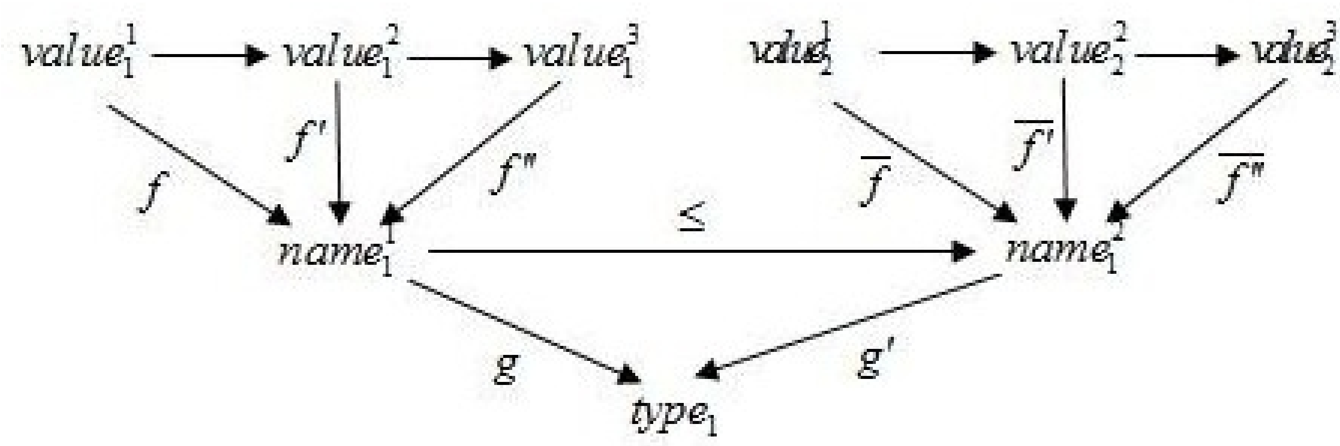
$$
\begin{tikzcd}
value_1^1 \arrow[r, "\eta_{value_1^1}"] \arrow[dr, "f"'] & Gname_1^1 \arrow[d, "Gg"] & \xrightarrow{F} & name_1^1 \arrow[d, "Ff"'] \arrow[dr, "g"] & \\
& name_1^1 & \xleftarrow[G]{} & Fname_1^1 \arrow[r, "\varepsilon_{type_1}"'] & type_1
\end{tikzcd}
$$

F --| G

η_{value}: *value'* → G(*name'*)

ε_{type} : FG(*name'*) → *type'*

Example Mapping derived from Relative Ordering of F --| G



Can think of adjointness as a relative ordering.
The contravariant mapping enables arrows *g* to *type* to be slotted onto the appropriate *name*.
name and *type* are both preorders.
Contravariant functors reverse the direction of composition.

Nature of Categories in Three-level Architecture

Consider categories DAT and SCH.

DAT holds values in the form:

value → *name*

But the basic structure must be more complex than this as also need to hold:

Relationships between values

Identifiers of data items

Suitable Constructions

Cartesian closed for connectivity, product (universal relationship) and identifier.

Locally cartesian closed to refine relationship from a general product to a specific context.

Locally cartesian closed categories:

- Pullback
- Comma (two functors with same codomain, Lawvere)

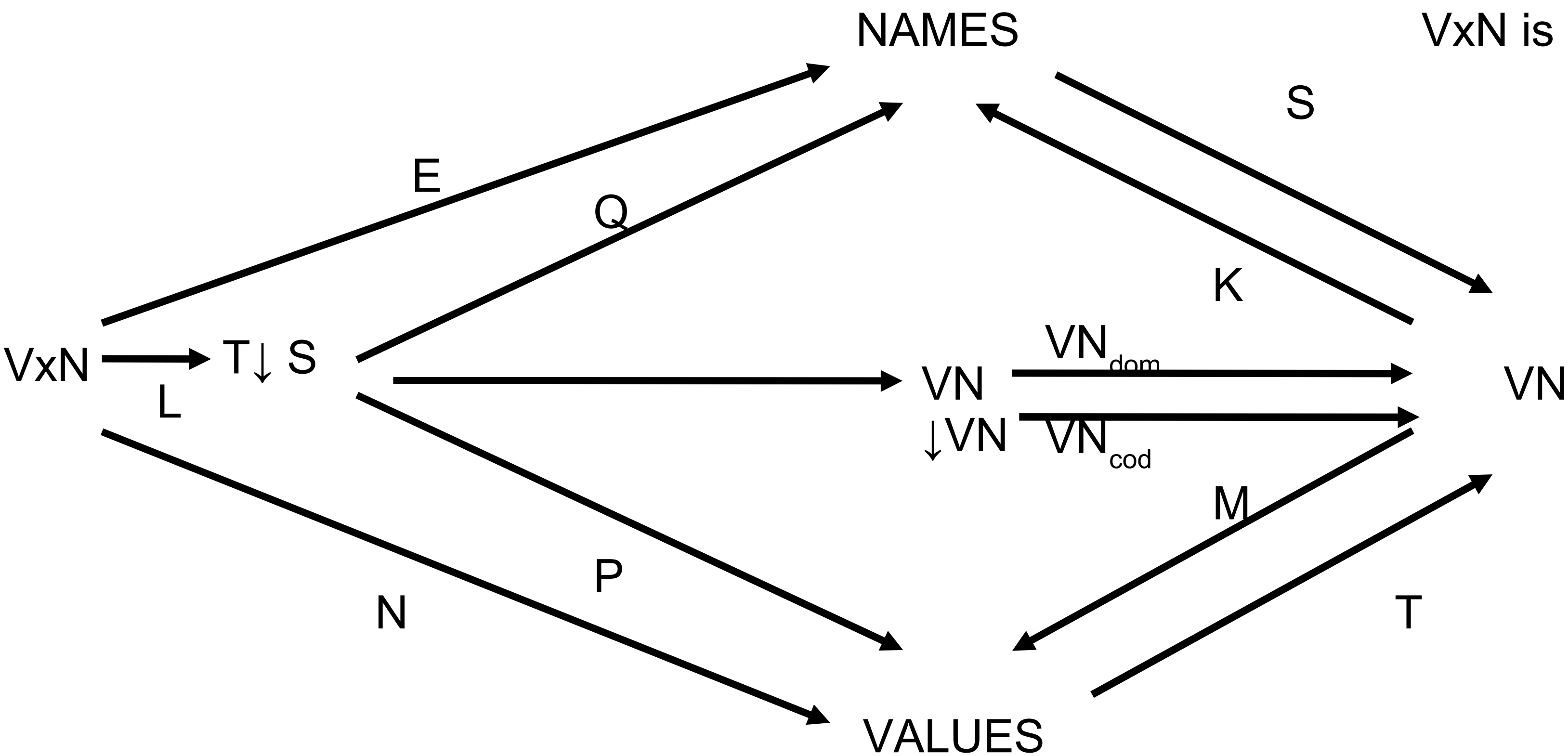
In theoretical computing much use of locally cartesian closed categories after work by Seely on type theory of Martin-Löf.

Adjoint Functors in a Pullback Diagram in level DAT

VN is NAMES+VALUES

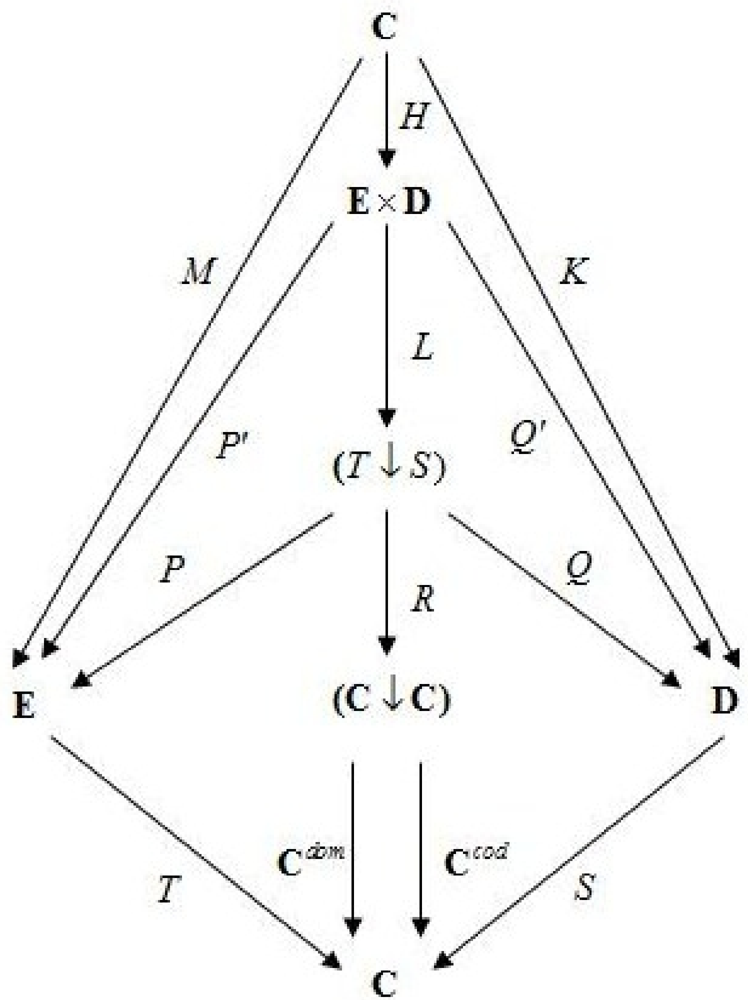
VxN is NAMESxVALUES

VN_{dom}, VN_{cod} are forgetful functors



T↓ S is comma category, relationship between T and S in context of VN;
T is identity functor on VALUES in VN, S is selection functor on NAMES in VN;
P, Q, E, N are projection functors; M is right adjoint to T; K is right adjoint to S; L is product functor

Generalisation of DAT



Comma category T↓ S acts as a constraint limit on the product category E x D

Product functor L and inclusion functor R compose a bifunctor (binary functor)

T may be free functor

S may be underlying functor

D and E can be any level in four levels of categories involved in three-level architecture

C is E+D

Further Work

- Investigate further the generalisation of DAT
  - Look at applicability to dynamic application of three-level architecture
  - Suggested application domain is security
- Explore further the details of DAT and SCH
  - Conditions for isomorphism

Contravariance and Anticipation

An anticipatory system is but the structured ordering of adjointness between the systems as a whole and every locality within it.

- Thus in the special case of time, the present is the particular locality of interest as a reductionist self-duality.

The past and the present is a contravariant view of the past from the present

- while the present access to future states is also a contravariant arrow from the future to the present.