

Sesiunea de primăvară a DLMFS: **Ce întemeiază știința, ce întemeiază tehnica**

Aprilie 2019

Experiență – creație – cunoaștere

GHEORGHE M. ȘTEFAN

Experiență – creație – cunoaștere

Tot ceea ce omul întemeiază devine inteligibil în complexul creat de **experiențele** pe care le trăiește și de reacțiile sale **creative** ce se revarsă în **cunoașterea** pe care o acumulează. Jocul dintre fenomen și formă în acțiune și gândire determină modurile întemeierilor pe care omul le realizează în lumile sale. Este vorba despre procese pe care le percepem disociate sub forma științei, tehnicii, matematicii, artei, Dar **distincțiile** existente nu justifică **disocierile** pe care modernitatea le-a impus. Există șansa unei echilibrate reintegrări care să păstreze distincțiile utile eliminând disocieri disfuncționale?

În universitățile medievale: cele 7 arte liberale

Trivium (soft skills):

- ▶ Grammar
- ▶ Logic
- ▶ Rhetoric

Quadrivium (hard skills):

- ▶ Arithmetic
- ▶ Music
- ▶ Geometry
- ▶ Astronomy

Modernitatea: *distincțiile => disjuncții*

- ▶ Modernitatea (> 1455) + Martin Luther (1517) declanșează procese de segregare acceleratoare
- ▶ Accelerarea proceselor de creație și cunoaștere cerea compartimentări simplificatoare
- ▶ Se pierde conexiuni pentru a se câștiga aprofundarea compartimentată



Oameni de știință

Ingineri

Matematicieni

Artiști

Medici

(Politicienii)

Juriști

Antreprenori

Educatori

Paznicii

...

Multitudinea modurilor întemeietoare

Știința: dezvăluie realitatea existențială

Tehnologia: instanțiază în realitatea lumii

Matematica: modelează realitatea mentalului uman

Arta: diversifică realitatea lumii & mentalului uman

Antreprenoriatul: gestionează realitatea

(Politica: perturbă realitatea)

...

Există o șansă pentru integrare?

- ▶ Mecanismul transdisciplinar profund în versiunea statică:
 - ▶ Experiență
 - ▶ Creație (processe noetice și noematice)
 - ▶ Cunoaștere
- ▶ Versiunea dinamică:
 - ▶ Experiență – Acțiune
 - ▶ Creație – Modelare interioară
 - ▶ Cunoaștere – Speculație

Tehnologie

Știință

Matematică

Educație

Antreprenoriat

Artă



Arhitectura ca mediator

Arhitectura: setul de funcții la interfața dintre două realități

- ▶ Fenomenul
- ▶ **Funcția** se interpune integrator
- ▶ Forma



Structurile și formele intră sub incidența
instrumentelor artificiale (IA, ...)

Mentalul uman se va concentra pe funcții evitând
detalii structural-formale

Gândirea arhitecturală va substitui travaliul
structural-formal

Omul devine astfel mai **liber** dar și mai **responsabil**

Complexitatea descrierii structural vs. simplitatea descrierii functionale

DESCRIERE STRUCTURALA

```
module dec3(output [7:0] out, input [2:0] in );
    wire in0, nin0, in1, nin1, in2, nin2;
    not not00(nin0, in[0]); not not01(in0, nin0) ;
    not not10(nin1, in[1]); not not11(in1, nin1) ;
    not not20(nin2, in[2]); not not21(in2, nin2) ;
    and and0(out[0], nin2, nin1, nin0); // output 0
    and and1(out[1], nin2, nin1, in0 ); // output 1
    and and2(out[2], nin2, in1, nin0); // output 2
    and and3(out[3], nin2, in1, in0 ); // output 3
    and and4(out[4], in2, nin1, nin0); // output 4
    and and5(out[5], in2, nin1, in0 ); // output 5
    and and6(out[6], in2, in1, nin0); // output 6
    and and7(out[7], in2, in1, in0 ); // output 7
endmodule
```

DESCRIERE FUNCTIONALA

```
module dec #(parameter inDim = 3)(input [inDim - 1:0] sel,
    output [(1 << inDim) - 1:0] out);
    assign out = 1 << sel;
endmodule
```



Thank you

Questions & possible answers

DESCRIERE STRUCTURALA

```
module dec3(output [7:0] out,
            input [2:0] in );
    wire in0, nin0, in1, nin1, in2, nin2;
    not not00(nin0, in[0]);
    not not01(in0, nin0) ;
    not not10(nin1, in[1]);
    not not11(in1, nin1) ;
    not not20(nin2, in[2]);
    not not21(in2, nin2) ;
    and and0(out[0], nin2, nin1, nin0); // output 0
    and and1(out[1], nin2, nin1, in0 ); // output 1
    and and2(out[2], nin2, in1, nin0); // output 2
    and and3(out[3], nin2, in1, in0 ); // output 3
    and and4(out[4], in2, nin1, nin0); // output 4
    and and5(out[5], in2, nin1, in0 ); // output 5
    and and6(out[6], in2, in1, nin0); // output 6
    and and7(out[7], in2, in1, in0 ); // output 7
endmodule
```

DESCRIERE FUNCTIONALA

```
module dec #(parameter inDim = n)
            (input [inDim - 1:0] sel,
             output [(1 << inDim) - 1:0] out);
    assign out = 1 << sel;
endmodule
```