On a Federated Architecture for Utilizing Domain-Specific Descriptive Models

F. Kamdem Simo*,[1,2] & D. Ernadote[2] & D. Lenne[1]

[1] Alliance Sorbonne Université, Université de Technologie de Compiègne, France

[2] Airbus Defence and Space, France

*frks@protonmail.ch

Context and objectives

A descriptive model is model whose structure — equipped with input and output interaction points — encodes the main (abstract) structure of the thing to be modelled or studied.

- Systems Engineering is different from System Design
- Clarity and precision are essential for model reuse, but not sufficient
- Architecture of descriptive models influences their definition and utilization
- Category Theory can be considered as a good architectural framework
- What is the fundamental organisation of a descriptive model ?
- How to fully take advantage of this organisation in practice ?

This work is not concerned with a modelling language, tool or methodology, but it is concerned with a federated architecture FA (unifying models on their structure and distinguishing them by interpretations of this structure) aimed to facilitating the utilization (specification, analysis, share, reuse, etc.) of models without sacrificing clarity and precision.

We study, not the "real" systems, but the system: a descriptive model.

Considering metamodels, data formats, interfaces, mappings etc.: to what extent can descriptive models be decoupled from the logic of their implementation by a particular tool (or source) while retaining their full meaning and being reusable in another tool (or place) by different actors.

- How to ensure that the implementation reflects the (modelling) formalism or the mathematical object ?
- Different implementations coexist
- The only solution might be to use the same implementation – like software libraries, yet...
- Mappings can help connect models, but managing mappings can raise the same issues as with models

Main idea: encapsulation-differentiation

The parts of the architecture of a descriptive model can be:

- **Structure of a model** This structure consists of boxes linked with incoming and outgoing wires. Input and output interaction points are ports associated to boxes ensuring connection with their surrounding environment. A representation of a tricky system can be managed using hierarchical decompositions and re-compositions.
- **Interpretations of the structure** Many systems are usually studied from different perspectives, namely: human, physical, structure, behaviour, cost, safety, etc., then different interpretations of their structure(s) are possible and necessary. At the level of models, these interpretations yield domain-specific languages or semantics.
- **Instances of the structure** Utilizing a structure involves data relating to an actual (or candidate) system that is modelled. These data should also relate to an interpretation of the structure, meaning that their precise meaning can be known and that they can be utilized.

Why Cateory Theory ? This framework enables us to

- naturally define the notion of composite-box with its constituent-boxes via an arrow in a category
- abstract away details relating to a particular interpretation of the structure of models
- define a formal link to be specified between the structure of models and various targets like Instances and Interpretations. This formal link is ideally defined as a functor.
- consider possible canonical extensions of FA with the enrichment of objects and category structure

We define the structure of these models as a symmetric multicategory. In particular, we rely on matrices over a semiring to define morphisms and their composition. The choice of matrices is intended to facilitate the application in practice.

Components of FA

Structure of a descriptive model

A (directed) Wiring Diagram WD [4] [7] looks like Figure 1, but in fact it differs from a WD in the following respects. We allow:

- Unconnected ports (ports without incoming or outgoing wires)
- Converging wires (separate wires with the same target port)
- Several links between a source port and a target port

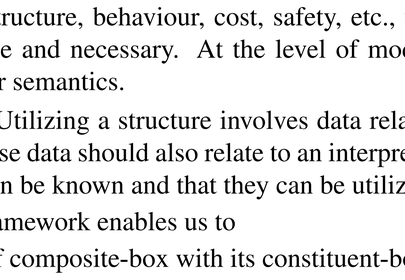

Figure 1: A box Y composed of 2 boxes X_1 and X_2

The point of allowing these various things is to encompass as many wiring patterns as possible that may occur in practice. For instance, it might be the case, structurally, that an actual component is not plugged into its surrounding environment via any of its interfaces. See for instance a simple Ptolemy model [6, Figure 9], similarly permissive.

Let K be the power set of the set of strings of finite lengths. $(K, \cup, \cdot, \emptyset, \{\epsilon\})$ is a semiring [1] where $\times := \cdot$ is the product induced by the string concatenation operator, $+ := \cup$ is the addition given by union of sets of strings of finite lengths, $0 := \emptyset$ is the zero given by the empty set, and $1 := \{\epsilon\}$ is the unit given by the singleton set containing the empty string.

Definition 0.1. MatMsc$_K$ is an underlying symmetric multicategory of (**Mscm**,$\otimes$) defined in the full companion paper.

- Its objects are the same objects as **Mscm**. An object X is rewritten as (X^{in}, X^{out}).
- Its arrows **MatMsc**$(a_1, ..., a_n; a)$. An arrow $a_1, ..., a_n \to a$ is rewritten as follows. Let $b = a_1 \otimes \cdots \otimes a_n$, $X^{in} = in(b)$, $X^{out} = out(b)$, $Y^{in} = in(a)$ and $Y^{out} = out(a)$, the arrow $b \to a$ in **Mscm**, noted $X \to Y$ is given by the couple of matrices (M^{in}, M^{out}) where $M^{in}$and M^{out} are given by the functions $X^{in} \times (Y^{in} \cup X^{out}) \to K$ and $Y^{out} \times X^{out} \to K$ respectively. Objects and arrows are subject to R1, R2 and R3 as in **Mscm**.
- Identities **MatMsc**$(a; a)$. For any object (X^{in}, X^{out}), the identity arrow $1_X := (1^{in}_X : X^{in} \times (X^{in} \cup X^{out}) \to K, 1^{out}_X : X^{out} \times X^{out} \to K)$, rewritten as follows. 1^{in}_X and 1^{out}_X yield $\begin{matrix} & X^{in} & X^{out} \\ X^{in} & \mathbb{I}_M & \emptyset_M \end{matrix}$ and $\begin{matrix} & X^{out} \\ X^{out} & \mathbb{I}_M \end{matrix}$

- The composition formula is the same as for **Mscm**, but rewritten as follows. Consider the objects X, Y, Z and arrows $n : X \to Y$ and $m : Y \to Z$ given by matrices (N^{in}, N^{out}) and (M^{in}, M^{out}) respectively. The arrow $m \circ n : X \to Z$ is given by (O^{in}, O^{out}):

$$\begin{aligned} O^{in} &= N^{in} \times M'^{in} \times N'^{out} \\ O^{out} &= M^{out} \times N^{out} \end{aligned}$$

$$\begin{array}{ccc} X^{in} & \overset{O^{in}}{\dashrightarrow} & Z^{in} \cup X^{out} \\ {\scriptstyle N_{in}}\downarrow & & \uparrow{\scriptstyle N'^{out}} \\ Y^{in} \cup X^{out} & \xrightarrow[M'^{in}]{} & Z^{in} \cup Y^{out} \cup X^{out} \end{array} \qquad \begin{array}{ccc} Z^{out} & \overset{O^{out}}{\dashrightarrow} & X^{out} \\ {\scriptstyle M^{out}}\downarrow & \nearrow_{N^{out}} & \\ Y^{out} & & \end{array}$$

Composition and associativity laws follow from matrix multiplication. It is worth noting that with only **MatMsc** (or **Mscm**), no meaning (related to a domain of interest or business domain) is assumed for an object (or interface), its ports and, if applicable, its internal links.

Formal link with Wiring Diagrams An arrow $b \xrightarrow{\theta} a$ in **Mscm**$(b; a)$ indicates how a box a is internally built from the box b. It consists of a tuple $(\theta^{in}, \theta^{out})$

$$\begin{aligned} \theta^{in} &: L^{in} \to in(b) \times (out(b) \cup in(a)) \\ \theta^{out} &: L^{out} \to out(a) \times out(b) \end{aligned} \tag{1}$$

where L^{in} and L^{out} are the abstract finite sets of links coming respectively into an input port and an output port of one of the boxes b and a. We also demand R1, R2, R3 available in the full paper.

A WD is basically given by

$$(X^{in} + X^{out}) \to (Y^{in} + X^{out}) \leftarrow (Y^{in} + Y^{out}) \tag{2}$$

which can be decomposed into two functions

$$\begin{aligned} \phi^{in} &: X^{in} \to Y^{in} + X^{out} \\ \phi^{out} &: Y^{out} \to X^{out} \end{aligned} \tag{3}$$

To obtain (1) – our definition – from (3), the elements of ϕ^{in} and ϕ^{out} must be associated to the abstract sets of links or wires L^{in} and L^{out}. To get (3) from (1) it is necessary to have θ^{in} and θ^{out} of (1) that are injective. The elements of L^{in} and L^{out} can be seen as the labels of wires.

Interpretations and Instances of the structure

Interpretations Using the structure of models, it is possible to specify an interpretation of boxes such that the associated semantics can be recovered. An actual interpretation could therefore be independently developed by using the structure of models. Ideally, the specification of the interpretation is given by a functor F : **MatMsc** $\to$ **Msin**, **Msin** being the category in which the structure is interpreted. The caveat 'ideally' is necessary, because it may not always be straightforward to define a functorial semantics. The concepts defined in **Msin** must be exposed by F, irrespective of how these concepts are defined in **Msin** and irrespective of the implementation. These data must be machine-accessible. Functorial semantics of wiring diagrams (i.e. special cases of **MatMsc**) are given in [3], [4], [7], ...

Instances If the structure of a model is to be automatically exchangeable, then in the same way that the definition of an interpretation of the structure needs to be machine-accessible, the related data must also be machine-accessible.

The data relating to an actual (or candidate) system modelled is defined by a Set-valued functor I : **MatMsc** $\to$ **Set**.

It can be argued that the effective data related to **MatMsc** should instead be given by a pushout over

$$\textbf{Msin} \xleftarrow{F} \textbf{MatMsc} \xrightarrow{I} \textbf{Set} \tag{4}$$

Conclusion

See the full paper for more details on related work and technical details. One perspective is to address how models evolve (structurally) over time [2].

References

[1] Manfred Droste & Werner Kuich (2009): *Semirings and Formal Power Series*, pp. 3–28. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-01492-5_1.

[2] Andrée C Ehresmann & J-P Vanbremeersch (1987): *Hierarchical Evolutive Systems: A mathematical model for complex systems*. *Bulletin of Mathematical Biology* 49(1), pp. 13–50.

[3] Dylan Rupel & David I Spivak (2013): *The operad of temporal wiring diagrams: formalizing a graphical language for discrete-time processes*. *arXiv preprint arXiv:1307.6894*.

[4] David I Spivak (2013): *The operad of wiring diagrams: Formalizing a graphical language for databases, recursion, and plug-and-play circuits*. *arXiv preprint arXiv:1305.0297*.

[5] David I Spivak (2015): *Nesting of dynamic systems and mode-dependent networks*. *arXiv preprint arXiv:1502.07380*.

[6] Stavros Tripakis, Christos Stergiou, Chris Shaver & Edward A Lee (2013): *A modular formal semantics for Ptolemy*. *Mathematical Structures in Computer Science* 23(04), pp. 834–881.

[7] Dmitry Vagner, David I Spivak & Eugene Lerman (2015): *Algebras of open dynamical systems on the operad of wiring diagrams*. *Theory and Applications of Categories* 30(51), pp. 1793–1822.

Acknowledgements

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02