**20TH INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2018**

TRIESTE, ITALY, OCTOBER 15 – 17, 2018

# The Algebra, Logic and Topology of System-of-Systems

Mats Johansson[1], Patrik Eklund[1], Jari Kortelainen[2], Michael Winter[3]

[1] Umeå University, Department of Computing Science, Umeå, Sweden
[2] South-Eastern Finland University of Applied Sciences, Mikkeli, Finland
[3] Brock University, Department of Computer Science, St. Catharines, Ontario, Canada

**Abstract:** In this paper we propose an information structure enrichment of relational models underlying design structure models typically used in System-of-Systems. Such design structures are algebraically, logically and topologically mostly unstructured relations as treated within naïve set theory. The paper also aims to show how an enriched information structure can be applied to monitor the health status of a System-of-System as an alternative to fault trees.

*Keywords: Contact lattices, generalized terms, nearness, proximities.*

## 1 Introduction

System-of-Systems embrace several intertwined subsystems and even several interrelated subsystem models involving humans and machines, physics and economics, evaluations and predictions, and many more aspects, all having specific modelling requirements. Systems are designed and manufactured, operated and maintained, and eventually replaced. From a system point of view, and while operational, the lifespan of a subsystem involves condition monitoring, identification of changes, and various aspects and phenomena that needs to be quantified and qualified, often in stochastic and many-valued settings. Monitoring of operations often involves identifying or preventing defect, as a matter of diagnostics. On the other hand, system functioning is important to maintain at required levels, or restored after shutdown or breakdown. Service and maintenance therefore has to focus both diagnostics as well as functioning.

As an example, any system that includes running mechanical components is affected by wear. In many cases, there are predictive models describing the effects of this wear over time and these models are the base for maintenance schedules. In some cases the actual wear of individual components will deviate from the predictive model and in these cases it is useful to have a system that may detect this deviation. Many systems are equipped with different kinds of sensors. In a system-of-systems there may be a number of predictive models that may or may not be similar in kind. There might be models for mechanical wear, models for fluids, air filters etc. Each will contribute to a general representation of the current projected status of a system-of-systems. Many systems are equipped with different kinds of sensors that may be used to detect deviations from the predicted models, to complement the models and to give a better general representation of the system status and not to forget they may be used to make new and better prediction models for maintenance of system and system-of-systems. This means that there is a need to be able to handle sets of data from different models that all aims to express various forms of states, but that are not necessarily using the same terminology. The way in which they differ might be expressed as a distance or rather *nearness*, since they are related in some way. Since a system or a system-of-systems by necessity is contributed by more than one individual part there will most likely be individual aspects that needs to be addressed that may or may not have a high level of nearness from a topological point of view.

To explain this we start with one of the simplest mechanical systems possible. Two gears who's cogs are linking in to each other. They have the exact same amount of cogs, i.e. a 1:1 ratio and are suspended in mid air without any bearings or axels. In this case any observation or prediction model for any of the two gears would be highly identical with the other. Since each cog would touch another cog the same amount of times the wear would be very similar regardless of which gear is chosen for observation or model. The nearness between the sets of terms created would be very close. Anything made to increase the complexity of this simple system will introduce differences between observations made, even if we use the same basic model.  Say that we change the ratios between the gears to 2:1. This would mean that each individual cog on one gear will touch a cog on the other gear twice as often during a finite amount of time. This would mean that even if it would be quite possible to use the same basic model to predict the wear of each wheel, we also will have to make and introduce a new model that handles the combined wear of both gears since the increased wear on one gear in fact may affect the other gear as a consequence. In this case it is easy to see the relation, or nearness in the cause and effect between the components regardless of how it is expressed since it is a very small system. If this is scaled up in to a system-of-systems the importance of the use of nearness as a way to express relation is far more important. Since a system-of-systems with high probability will be made using components of different makes and vendors using different kinds of standardizations or even vendor specific notations for diagnostics. There will be a need to express how closely related seemingly different values or terms are. Both to draw conclusions about the current status of a system-of-systems but also to identify unobvious relations that may enable better conclusions about the overall state of a system-of-systems.

Most mechanical devices, regardless of the existence of electrical components, can be viewed as a singular system or a system-of -systems. A gearbox may be seen either as a system for changing gear ratios, as a part of a transmission system, as a part of a drive system or similar. Even if the gearbox is viewed as a singular system it may still be possible to divide it into a functional part, the gears, and an enabling part, the bearings, and if present even to a controlling part, a gear selector. Loss of function in any individual subsystem will probably reduce the overall function of the gearbox, but not necessarily make it inoperable. Should the gearbox be viewed as part of a transmission system the problem becomes more complex making the need for a more sophisticated logic for accurate diagnostic. Damages to peripheral parts of the transmission system might increase the wear on individual gears making their predicted wear inaccurate. In respect, problems originating in the gearbox might lead to increased wear on things like bearings and motors but not necessarily stop the system ability to operate as a whole. In other words, the system or system of systems experiences a loss of function and needs a correct diagnosis to determine a correct cause of action, but the system has not stopped working.

If there was to be an analogy with the human body, a machine would not only be considered either operational or non-operational, it would be considered as either healthy or affected by different levels of function loss. A person suffering from a arthritis in a thumb joint would not be considered non-operational. That person might even be considered quite healthy over-all. The idea is that the amount of wear on a mechanical subsystem, or even a fully mechanical system, cannot be represented by either a 0 (false) or 1 (true), or possibly even a scale e.g. from one to five. Our point in this paper is then also that this is not just a numerical scale, but comes with algebraic structures. It needs to be translated into a much more sophisticated representation to fully represent the complexity of the problem. A classical logical fault-tree consisting of either true or false as possible states of being are not accurate enough even for a small system. A system or system-of-system that could come in question for scrutiny of its dependencies and structures must be equal to a process. There would be little need to perform such task on a static object. In order to sufficiently translate the overall health state of a system or a system-of-system we need to use generalized relations and logical models that allows for order and many-valuedness. This means that the classical fault tree, that uses 0 and 1 to represent operational or non-operational states is replaced by something that is containing enriched information, perhaps in the form of truth values between $bot$ (bottommost truth value) and $top$ (topmost truth value), enabling representation of the operational degree of any given system.

One fundamental aspect of applying any form of algebraic, logic or topological operation on a system of even moderate complexity is to have the means of understanding a real-life-system with its interactions, both internal and external, and to have a tool capable of making a logically coherent visualization of this system. There are a number of established notations available that are more or less widely used to translate different kinds of processes in to structured and ordered representations, or models, of the original. The more complex the system and the more intricate the system-of-system, the higher is the need to find a notation with a rich underlying logic. This is important to allow for design structures to keep relations between the components and data and allow for maintaining both order, many-valuedness and topology.

## 2 Unstructured and structured information

The simplest form of information is a set $X$ of points $x \epsilon X$. If $X$ is given no structure, and the points $x$ remain unexplained, no mathematics, apart from set theory, can be applied to analyze such ‘information’.

Intuitively, we may e.g. say that $X_{Co}$ is a ‘set of components’ and $x_{crankshaft}$ is a ‘component’ in $X_{Co}$, i.e., $x_{crankshaft} \epsilon X_{Co}$. It is then tempting to say that this is more informative than saying $x \epsilon X$, but in fact, mathematics at this point is blind to see any difference between $x_{crankshaft} \epsilon X_C$ and $x \epsilon X$, since $x_{crankshaft}$ is mathematically still just an element and $X_{Co}$ is just a set.

The DSM model (Eppinger and Browning, 2012) is a typical relational model, which informally may define information types, and in the case of DSM roughly divide these types into *components*, *people* and *activities*. Respective types are equipped with underlying and unstructured sets of elements of these types, so that we may add sets $X_{Co}$, $X_{Pe}$ and $X_{Ac}$, respectively, of elements representing components, people and activities. However, elements in these sets indeed remain simply as names. Algebraically, logically and topologically we still have very little structure, if any structure at all, except for the possibility to create free algebras, logical signatures with only constants, or trivial topologies.

A typical step and starting point to add structure is to say that "points can be related". We may want to describe how components are related or maybe how components and people are related, and so on. This means we establish relations as subsets

$$R_{CoCo} \subseteq X_{Co} \times X_{Co}$$

and

$$R_{CoPe} \subseteq X_{Co} \times X_{Pe}.$$

We may want to impose various properties on relations, like those for reflexivity, symmetry and transitivity, providing equivalence relations. Such relations divide the set of elements into a set of non-overlapping subsets. Conversely, for any subdivision of a set into a set of non-overlapping subsets we can define a unique equivalence relation that provides that subdivision. Respective subsets are then *per se* unrelated.

The symmetry property essentially means that the relation is unordered, so that asymmetry means that order makes sense. The relation is then more conveniently treated as an order relation, and therefore appears within the realm of lattices and algebras.

Note also how a relation $R \subseteq X \times X$ can be equivalently represented as the mapping

$$\rho : X \times X \to 2$$

where 2 denotes the two-pointed set $\{0,1\}$ $(or\ \{false, true\})$. The relation has initially no properties, so it may e.g. be asymmetric indicating that the order between components is important. However, order as a structure is not explicitly recognized within the formal notation, and in fact, in the case of DSM, the model comes with very little formal notation.

In design structures, order and many-valuedness are important, but in logic it is an interesting question whether order precedes many-valuedness. If we first extend 2 to $Q$, a non-commutative quantale, we have a many-valued relation

$$\rho : X \times X \to Q$$

and non-commutativity of the quantale means that aggregations will consider the order among elements in $Q$, see e.g. (Eklund, Gutiérrez García, Höhle and Kortelainen, 2018). DSM also deals with many-valuedness, but in a rather pragmatic way, and not using algebraic notions or logical formalism to describe it more precisely.

This is clearly seen e.g. in DSM's four types of interactions (spatial, energy, information, and materials), with a 5-scale (-2 … 2) characterizing many-valuedness for each interaction. That 5-scale can be viewed as a quantale, but the relation between respective 5-scales is not algebraically explained in DSM.

Many-valuedness and order is thus poorly explained in DSM, and for the set $X$ must also have a more elaborate structure, otherwise the size of that unstructured set quickly grows to become very large, and application development makes no practical sense. As we indicated before, $X$ cannot be just a *set of* elements. It has to be a *structure of* elements.

As an example, if we only say ‘crankshaft’ as a name for a component in an automotive system-of-systems, ‘crankshaft’ is just a logical constant, but if we include the attributes $attr_1, \dots, attr_n$ attached to a crankshaft it becomes a logical term. Using logical notation, $crankshaft$ is a logical constant (of zero arity), whereas $crankshaft(attr_1, \dots, attr_n)$ is a term, with $crankshaft: s_1 \times \dots \times s_n \to s$ being an operator (of arity $n$) and $s_i$, $i = 1, \dots, n$, and $s$ are types (sorts).

In first order logic, $crankshaft(attr_1, \dots, attr_n)$ may be viewed as a term or a predicate. In (Eklund, Höhle and Kortelainen, 2014) terms are clearly separated from sentences, so that $crankshaft(attr_1, \dots, attr_n)$ is an expression (term) rather than a statement or predicate (sentence). Conglomerates of sentences become part of the logical *theory* related with the design structure.

In the simplest case, components are terms, built upon a signature $\Sigma = (S, \Omega)$, where $S$ is the set of types and $\Omega$ is the set of operators. The set of all terms (expressions) is then $T_\Sigma X$, where $X$ is a set of variables. The design structure is then

$$\rho : T_\Sigma X \times T_\Sigma X \to Q$$

where order and many-valuedness reside in both components and the valuation of the relation between them. In this situation, $T$ is a functor over the category of sets, so that order and many-valuedness reside in the functor structure. However, as explained in (Eklund, Galán, Helgesson and Kortelainen, 2014), $T$ can more generally be an endofunctor over any monoidal biclosed category, so that order and uncertainty is modeled in the underlying category (metalanguage) rather than in the functor itself.

Further, the relation $\rho$ may be constrained by properties, such as associativity. Applications typically define these properties, as well as the nature of order and many-valuedness.

We can enrich $\rho$ even further, and this makes us realize how DSM without structure is capable of producing applications on a very general level only.

## 3 Contact relations

People, and people in teams, are obviously differently structured as compared to components and subsystems of components. Relations between and (topological) nearness of people and teams require to be modelled also involving topological notions like neighbourhood, entourage, proximity and nearness. Neighbourhoods of points in topological models originate and abstracts from geometry and metric space models. Entourages in uniform spaces (Weil, 1937) and can intuitively be viewed as two-dimensional or “relational” neighbourhoods. Nearness (Herrlich, 1974) extends proximities (Riesz, 1909), where these models consider proximity of sets rather than points. This brings proximity consideration closer to the notion of *contact relations*.

The mathematical notion of contact has its origin in the so called point-free approach to topology. In recent years, point-free descriptions, i.e., region-based theories of space, in particular, have been a prominent area of research. Traditionally, space has been considered in mathematics by point-based theories such as geometric (e.g. Euclidean geometry) or topological representations (point-set topology) of space. Representing a region by the set of its points might be impossible or at least very inefficient when it comes to computer applications. As an alternative point-free theories of space such as region-based theories can be used to represent space in the context of qualitative spatial reasoning. Using regions instead of points as basic entities accounts more naturally for how humans conceptualize the physical world. For this reason this alternative representation of spatial entities and their relationships has become a prominent area of research within AI and Knowledge Representation. Since the earliest work of de Laguna (deLaguna, 1922) and Whitehead (Whitehead, 1929), mereotopology has been considered for building point-free theories of space. Mereotopology is a combination of the topological notion of connectedness with the mereological notion of parthood. A common mereological approach is to use Boolean algebras modeling the parthood relationship of regions. A Boolean algebra is a set $B$ with two binary operations $\wedge,\vee$, a unary operation $^*$ and two constants 0,1 so that the following axioms are satisfied:

| | | |
|---|---|---|
| $a \vee (b \vee c) = (a \vee b) \vee c$ | $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ | associativity |
| $a \vee b = b \vee a$ | $a \wedge b = b \wedge a$ | commutativity |
| $a \vee (a \wedge b) = a$ | $a \wedge (a \vee b) = a$ | absorption |
| $a \vee 0 = a$ | $a \wedge 1 = a$ | identity |
| $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$ | $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$ | distributivity |
| $a \vee a^* = 1$ | $a \wedge a^* = 0$ | complements |

With $a \leq b$ iff $a \wedge b = a$ the induced order on $B$ is defined that immediately generalizes the inclusion of set of points to the abstract elements of the Boolean algebra. A so-called contact relation is often used to model the topological aspect of regions of being in contact. Formally, a contact relation $C \subseteq B \times B$ is a binary relation on $B$. Most commonly, the following axioms for $C$ are considered:

| | | |
|---|---|---|
| $C0$ | $\neg(0Ca)$ | null disconnectedness |
| $C1$ | $a \neq 0 \rightarrow aCa$ | reflexivity |
| $C2$ | $aCb \rightarrow bCa$ | symmetry |
| $C3$ | $aCb$ and $a \leq c \rightarrow aCc$ | compatibility |
| $C4$ | $aC(b \vee c) \rightarrow aCb$ or $aCb$ | summation axiom |
| $C5$ | $C(a) = C(b) \rightarrow a = b$ | extensionality |
| $C6$ | $aCc$ or $bCc^* \rightarrow aCb$ | interpolation axiom |
| $C7$ | $a \neq 0$ and $a \neq 1 \rightarrow aCa^*$ | connection axiom |

The first axiom says that no region is in contact with the empty region and $C1$ requires that every non-empty region is in contact to itself. The symmetry axiom makes contact a symmetric relation. This axiom makes perfectly sense in the spatial interpretation. However, if we consider parts of an engine or a system and interpret contact to model the potential influence of a mail function in one part on the other part, this axiom might not be suitable. $C3$ relates the order structure, i.e., the mereological notion, to the notion of contact. The summation axiom states that if a component $a$ is in contact to a component that consists of two parts, then $a$ must be in contact to at least one of the parts. The extensionality property ties the mereological notion to contact. It requires that if to components are contact to the same set of parts, then they are equal. As a consequence the order relation becomes definable in terms of $C$. The interpolation axiom is an axiom that stems from contact relations obtained by proximity spaces. It is a separation property requiring that two disconnected regions, i.e., two regions that are not in contact, there is a third region disconnected from the first including the second as non-tangential part. Finally, $C7$ requires that every non-trivial region is connected to its complement.

Please note that Boolean contact algebras, i.e., Boolean algebras together with a contact relation satisfying $C1 - C4$, can be represented in topological spaces with the usual definition of contact. In this context the additional axiom correspond to certain properties of the topological space.

## 4 The Information & Process view of relational structures

In order to translate real world systems into some equivalent representation that can be manipulated and interpreted, some kind of transitional layer is needed. Careful use of BPMN or DMN to capture a real-world process may both preserve and reveal relations between active components in a logically consistent way. Tools like BPMN can be used to make representations of many things and system of systems are just one example outside the business world. Since BPMN and its siblings allows for dependencies like directional flows and relations the addition of weights and values makes them well suited to apply logic to allow for better ways to understand the inner workings of any system of systems, they do however have limitations.

In (Eklund, Johansson, Kortelainen and Salminen, 2017) the logically extended view of DSM was promoted with respect to design structure becoming potentially supported by *information and process* standards as appearing in the OMG (Object Management Group) family of languages and notations, including

- UML (Unified Modeling Language)
- SysML (Systems Modeling language)
- BPMN (Business Process Modeling Notation)
- CMMN (Case Management Model and Notation)
- DMN (Decision Model and Notation)

UML’s Structure Diagram is a database model, whereas the Behaviour Diagram in UML is less recognized and used. The Behaviour Diagram in fact is a process model. Further, UML’s Behaviour Diagram is part of SysML, which is a process model expanding the process model side of UML. SysML is intended e.g. to support systems-of-systems modeling in engineering and manufacturing. BPMN in OMG should not be confused with value chain models, and and the logic of DMN is basically a propositional logic on a very trivial and basic level. Systems-of-Systems indeed embrace UML, SysML, BPMN, CMMN and DMN, in a variety of combinations.

## References

P. Eklund, P., Johansson, M., Kortelainen, J., Salminen, V., 2017. The logic of DSM, In: Understand, Innovate, and Manage your Complex System!, Proceedings of the 19th International DSM Conference (Eds. K. Hölttä-Otto, T. R. Browning, S. D. Eppinger, L. Becerril), Espoo (Finland), 11–13 September 2017, 25-31.

de Laguna, T. (1922). Point, line and surface as sets of solids. The Journal of Philosophy 19, 449–461.

Eklund, P., Galán, M.A., Helgesson, R., Kortelainen, J., 2014. Fuzzy terms, Fuzzy Sets and Systems 256, 211-235.

Eklund, P., Gutiérrez García, J., Höhle, U., Kortelainen, J., 2018. Semigroups in Complete Lattices: Quantales, Modules and Related Topics, Developments in Mathematics 54, Springer, 2018.

Eppinger, S.D., Browning. T.R., 2012. Engineering Systems: Design Matrix Methods and Applications, MIT Press.

Herrlich, H., 1974. A concept of nearness, General Topology and its Applications 5, 191-212.

Riesz, F., 1909. Stetigkeit und abstrakte Mengenlehre, Rom. 4. Math. Kongr. 2: 18–24.

Weil, A., 1937. Sur les espaces à structure uniforme et sur la topologie générale, Act. Sci. Ind. 551, Paris.

Whitehead, A. N, 1929. Process and reality. MacMillan, New York.

**Contact: M. Johansson,** Umeå University, Department of Computing Science, Umeå, Sweden, +46 90 786 6507, matsj@cs.umu.se