21st INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2019
MONTEREY, CA, USA, 23 – 25 September, 2019

Many-Valued and Many-Sorted Structure Relations

Patrik Eklund[1], Mats Johansson[1], Magnus Löfstrand[2], Michael Winter[3]

[1]Umeå University, Department of Computing Science, Umeå, Sweden
[2]Örebro University, School of Science and Technology, Örebro, Sweden
[3]Brock University, Department of Computer Science, St. Catharines, Ontario, Canada

Abstract: System-of-systems engineering and related models of engineering design require intertwining of structures, respectively, at least for components in products, engineering and business activities, and people involved in those activities as also specifically connected with production and its management. In this paper we describe the logical machinery of system-of-systems engineering and related models of engineering design in sufficient so as to enable to describe the algebraic foundation of the many-valued logic that is inherent in these systems and structures. We are thus essentially unravelling the hidden and underlying logic of these systems and their related information and process structures, focusing on interaction between elements of the system.

Keywords: Information, many-valued logic, process, system-of-systems

1 Introduction

System-of-systems engineering and related models of engineering design require intertwining of structures, respectively, at least for components in products, engineering and business activities, and people involved in those activities as also specifically connected with production and its management.

Products and components representing the ‘what’ of this complex system, people are ‘who’, and activities roughly ‘how’. The ‘where’ and ‘why’ add further complexity to the overall view of system-of-systems engineering and related models of engineering design. The ‘what’ and ‘who’ relates more to *information*, whereas ‘where’ and ‘how’ is more related to production and business *processes*. The complementing ‘why’ provides related logical statements and criteria based on ‘what’ and ‘who’ and as situated within ‘where’ and ‘how’. Logically speaking, ‘what’ (products and components) and ‘who’ (people) are detailed by their features and attributes, whereas ‘how’ and ‘why’ is supported by a guideline based on sets and structures of logical rules.

Information is either generally valued or truth valued. General values include numbers, ordinals and symbols, as subjected to functional (non-logical) operation, whereas truth values appear as binary or many-valued as subjected to logical operation.

In this paper we describe the logical machinery of system-of-systems engineering and related models of engineering design in sufficient so as to enable to describe the algebraic foundation of the many-valued logic that is inherent in these systems and structures. We are thus essentially unravelling the hidden and underlying logic of these systems and their related information and process structures.

2 Logic

In this section we explain our logical machinery in sufficient detail so as to enable to describe the algebraic foundation of our logical approach. Before outlining our structure of many-sorted terms, we should point out the importance of making a clear distinction between expressions (logical *terms*) and statements (logical *sentences* or formulae).

A logical expression, or a term, comes from an n-ary operator $\omega : \mathtt{s}_1 \times \ldots \times \mathtt{s}_n \rightarrow \mathtt{s}$, where $\mathtt{s}_1,\ldots, \mathtt{s}_n$ are the input types (sorts) of the operator, and $\mathtt{s}$ is the output type. If $t_1, \ldots, t_n$ are terms of respective types $\mathtt{s}_1,\ldots, \mathtt{s}_n$, denoted $t_i :: \mathtt{s}_i, i = 1, \ldots, n$, then $\omega(t_1, \ldots, t_n)$ is a term of type $\mathtt{s}$, denoted $\omega(t_1, \ldots, t_n) :: \mathtt{s}$. The output type may represent truth value, e.g., denoted as `bool`, so that we may have expressions like $S(x, y) :: \mathtt{bool}$. Such terms are, in standard first-order logic (predicate calculus), frequently called *propositions*. However, as a proposition, or statement, it is logically quite different from formulae like $\exists x. S(x, y)$. Standard first-order logic calls $S(x, y)$ and $\exists x. S(x, y)$ both formulas, but, in our approach to term constructions (Eklund, Galán, Helgesson and Kortelainen, 2014), $S(x, y)$ as a proposition is an 'expression' (term), provided by a term construction, whereas $\exists x. S(x, y)$ is a 'formula'. Note also how the basic design matrix in DSM (Eppinger and Browning, 2012) as a binary relation $R \subseteq X \times X$ can be equivalently represented as a mapping $\rho : X \times X \rightarrow 2$ where 2 denotes the two-pointed set $\{0,1\}$ (or $\{false, true\}$), i.e., representing binary (two-valued) truth. In this case, ρ can be seen as a candidate for a semantic interpretation of S in $S(x, y) :: \mathtt{bool}$, with $\{0,1\}$ being the semantic interpretation of the type `bool`.

In design structures, order and many-valuedness are important. In logic it is an interesting question whether order precedes many-valuedness, or vice versa. In (Eklund, Gutiérrez García, Höhle and Kortelainen, 2017) we argue that order underlies many-valuedness. We further show how category theory as a metalanguage, and monoidal closed categories in particular, underlies logical considerations related to order and many-valuedness. As pointed out in (Eklund, Johansson, Kortelainen and Salminen, 2017), if we extend this binary truth situation to many-valued logic, i.e., we extend 2 to Q, where Q is e.g. a non-commutative quantale (Eklund, Gutiérrez García, Höhle and Kortelainen, 2017), we have a many-valued relation $\rho : X \times X \rightarrow Q$ and non-commutativity of the quantale means that aggregations will consider the order among elements in Q.

In (Eppinger, Whitney, Smith and Gebala, 1994), a certain Q is introduced, when dealing with the issue of how to "document the technical interactions among the engineering parameters". In what is called the "Task-Level Design Description" there is potentially a $Q = \{Input, Feedback, Control, Addition\}$, but respective elements in Q are apparently not treated as qualifications of the same type, i.e., as a many-valuedness, but rather as a many-typedness of respective two-valued qualifications. Indeed, that Q is not assumed to have any algebraic structure. The complementing "Parameter-Level Design Description" and its qualification is two-valued only.

On the issue of hiding and unravelling information, note how the expression $\omega(t_1, \ldots, t_n)$ and its related value represents both the expression and the value. In the case of $2 + 2 = 4$, the value of the expression $2 + 2$ is 4. However, if we focus only on the value, we hide the expression. The value 4 is the value also for the expression $1 + 3$. So if we maintain the expression $2 + 2$, we can always compute its value, but if we only store the value 4, we cannot know which expression leads to that value. It might be tempting to introduce a variable x_ω to carry such values, and we might have $x_\omega = 4$. The clearly x_ω hides the expression $2 + 2$, or $\omega(t_1, \dots, t_n)$, in the general case.

We now look at a concrete example. For engine lubrication and engine oils, it is important that the oil is not contaminated. If it is, contamination is not binary, but is a degree. Oil may contain unwanted particles, and it may contain water. Both lead to contamination. If we would capture the contamination degree using a single variable

$$x_{oilContamination}$$

we will obviously hide valuable information provided by unwanted particles and water. On the other hand, if we have expressions for particle count and water amount, then

$$oilContamination\big(particleCount(filter), waterSensor(location)\big)$$

represents the value of oil contamination as computed based on particle count and amount of water in the oil, with values provided by suitable sensors, represented, respectively, by operators like

$$particleCount : \texttt{engID} \to \texttt{nat}$$

and

$$waterSensor : \texttt{engLocID} \to \texttt{num}$$

This is then based on the operators

$$oilContamination : \texttt{nat} \times \texttt{num} \to \texttt{bool}$$

$$particleCount : \texttt{engID} \to \texttt{nat}$$

$$waterSensor : \texttt{engLocID} \to \texttt{num}$$

$$filter : \ \to :\texttt{engID}$$

$$location : \ \to \texttt{engLocID}$$

The question "*Is oil contamined?*" is then a (many-valued logic truth) value of the expression

$$oilContamination\big(particleCount(Filter), waterSensor(Position)\big)$$

with the value of the expression being of type `bool`, i.e.,

$$oilContamination\big(particleCount(Filter), waterSensor(Position)\big) :: \texttt{bool}$$

In a simple and information hiding approach it is clearly easy to discretize to binary truth and use an expression free approach only using variables. We could have

$$x_{oilContamination} :: \texttt{bool}, x_{particleCount} :: \texttt{bool}, x_{waterSensor} :: \texttt{bool}$$

and simple use a proposition like

$$x_{oilContamination} = x_{particleCount} \ AND \ x_{waterSensor}$$

where AND: $\texttt{bool} \times \texttt{bool} \to \texttt{bool}$ corresponds to logical conjunction. In a binary situation, that AND is unique, but in many-valued situations, there are many alternatives for logical conjunction. Moreover, all such conjunctions need not be commutative. In the binary case we always have $x_1 \ AND \ x_2 = x_2 \ AND \ x_1$. In our example, we may ask which one is more contaminating, particles or water. In another example like $x_{suspension} \ AND \ x_{wheel}$ and $x_{wheel} \ AND \ x_{suspension}$ it is quite clear that they are not the same as failure in suspension is more likely affect the tires of the wheel than the other way around. If the conjunction AND in a many-valued setting is represented by the multiplication in a non-commutative quantale, then aggregation of ordered observations will be handled within that algebraic foundation represented by quantales.

Now adding e.g. $oilViscosity(\dots) :: \mathtt{bool}$ it is obvious how information is very much unraveled in a logical expression like

$$oilContamination(\dots)\ AND\ \ oilViscosity(\dots)$$

as compared to having binary truth in fault trees having propositional expressions like

$$(x_{particleCount}\ AND\ x_{waterSensor})\ AND\ x_{oilViscosity}$$

which hides counters and sensors, even if capable of e.g. providing risk values related to the functioning of the engine, but not being able to explain it, or locate the underlying reasons for elevated risk values. Note have this unravels a straightforward and information hiding approach where there would be a a single parameter-free variable like x_oilContamination to carry a value in a chosen scale for degrees of contaminations. Note also that the notion of 'Type' in (Pimmler, 1994) is intuitively comparable but not entirely equal to 'type' in our sense, i.e., as in many-sorted signature based logic and type theory.

3 The term construction

The category theoretical construction of one-sorted terms over a signature was given in (Eklund and Gähler, 1992) describing it as a term functor extendable to a monad, and doing so over the category $\mathtt{Set}$ of sets and functions. In (Eklund, Galán, Helgesson and Kortelainen, 2014) we extended the term functor and monad construction to be many-sorted and be applied over any monoidal closed category, including the Goguen category $\mathtt{Set}(Q)$, with Q being a quantale.

In this section we briefly outline the many-sorted term construction in order to be able to describe many-valuedness as appearing in the DSM matrix, to be described in section 4.

Terms are formed over a given signature $\Sigma = (S, \Omega)$, where S is a set of types (sorts), and Ω is a set of operators. The operator set may be many-valued, so that operators are attached with uncertainties, which leads to terms becoming attached with uncertainties. Informally we may denote the term functor over $\mathtt{Set}$ as $\mathtt{T}_\Sigma : \mathtt{Set} \to \mathtt{Set}$, not considering that we should use multi-sorted categories. For detail, see (Eklund, Galán, Helgesson and Kortelainen, 2014). An n-ary operator $\omega \in \Omega$ is denoted as $\omega : \mathtt{s}_1 \times \dots \times \mathtt{s}_n \to \mathtt{s}$, where $\mathtt{s}_1, \dots, \mathtt{s}_n, \mathtt{s} \in S$. Note that variables must now be sorted, so that a variable x must be connected with a particular type. The set of variables of type $\mathtt{s}$ is denoted $X_\mathtt{s}$, and the set of all terms of type $\mathtt{s}$ is denoted $\mathtt{T}_{\Sigma,\mathtt{s}}(X_\mathtt{s})_{\mathtt{s} \in S}$. For detail concerning the categorical term construction, see (Eklund and Gähler, 1992) and (Eklund, Galán, Helgesson and Kortelainen, 2014).

4 Interaction between elements

The "documentation of interaction between elements" as described in (Pimmler, 1994), and similarly in (Pimmler and Eppinger, 1994) is more of an informal and intuitive documentation than a formal and many-valued logical description of interaction. It is based on $\{Detrimental, Undesired, Indifferent, Desired, Required\}$, an ordered chain as a scale, equivalently annotated as a score $\{-2,-1,0,1,2\}$, and then further detailed, respectively, for Spatial Scale, Energy Scale, Information Scale and Materials Scale. For instance, in the case of Spatial Scale and Desired (+1), the verbal understanding of the scale value is “physical adjacency is beneficial, but not absolutely necessary for functionality”. It is indeed “necessary”, but not “absolutely necessary”. “Documentation” in the sense of (Pimmler, 1994) embraces 'function' descriptions for two related elements, like Radiator and Engine Fan, for which their 'relationship' is described. That relationship is then scored by a quadruple like $(+2,0,0,+2)$, i.e., the Spatial and Materials scores are $+2$, whereas the Energy and Information scores are 0.

More formally, this means there are four different relations on the set X of elements, where Radiator and Engine Fan are such elements, which can be denoted $\rho_{Spatial}$, ρ_{Energy}, $\rho_{Information}$ and $\rho_{Materials}$, respectively. In the example above, we would have

$$\rho_{Spatial}(Radiator, EngineFan) = +2$$

We should note that (Pimmler, 1994) and (Pimmler and Eppinger, 1994) do not classify or grade functioning as such. The functioning of the radiator is briefly described as “the radiator dissipates excess engine heat, via forced convection, to the outside surrounding”, where “forced convection” is obviously the key feature in that functioning. There is e.g. no ‘grade of convection’ or anything similar that would explicitly grade the functioning of the radiator.

Note also that there is no aggregation of the four scores into a common score of the relation between those two elements. We can introduce a tupled relation

$$\rho = (\rho_{Spatial}, \rho_{Energy}, \rho_{Information}, \rho_{Material})$$

so that we have

$$\rho(Radiator, EngineFan) = (+2,0,0,+2)$$

If we denote $\{-2,-1,0,1,2\}$ by L, and write E for the set of all elements, then ρ has the form $\rho : E \times E \to L^4$. Viewed in the framework of out term construction, we could see qualification annotated to the Radiator as the value of the expression $Radiator(\dots)$, being part of a term set $\mathrm{T}_{\Sigma,\mathrm{s}}(X_\mathrm{u})_{\mathrm{u}\in S}$ with T_Σ being a functor over $\mathtt{Set}$ rather than over $\mathtt{Set}(Q)$, which enables to invoke many-valued grading of expressions. For instance, Q could be the three-valued “traffic light”, so that the functioning of a radiator would be classified not just either as functioning or not functioning, but sometimes being in a transition state between the two. Indeed, this option of additional grading structure is not included in (Pimmler, 1994). Nor is it included in (Eppinger and Pimmler, 1994), which is basically a summary or (Pimmler, 1994), or in (Browning, 2001).

We could now also enrich the (Pimmler, 1994) model and make the set $\{\mathtt{Spatial}, \mathtt{Energy}, \mathtt{Information}, \mathtt{Material}\}$ to become a subset of the set S of types in the underlying signature. Then $Radiator(\dots)$ is viewed as an expression, respectively, from Spatial, Energy, Information and Materials point of view. So, with $Radiator(\dots)$ and $EngineFan(\dots)$ as terms e.g. in $\mathrm{T}_{\Sigma,\mathtt{Spatial}}(X_\mathrm{u})_{\mathrm{u}\in S}$, we would need to introduce the relation $\rho_{Spatial}$ as

$$\rho_{Spatial} : \mathrm{T}_{\Sigma,\mathtt{Spatial}}(X_\mathrm{u})_{\mathrm{u}\in S} \times \mathrm{T}_{\Sigma,\mathtt{Spatial}}(X_\mathrm{u})_{\mathrm{u}\in S} \to L$$

Crossover between the types now immediately comes into play, but this aspects is not considered in (Pimmler, 1994), which has a focus on interaction between elements, rather than interaction between scores. In (Pimmler, 1994), Spatial, Energy, Information and Materials are all seen in light of functioning. The distinction between fault and functioning, as outlined in (Eklund and Löfstrand, 2016), is not made explicit. Instead of types $\{\texttt{Spatial}, \texttt{Energy}, \texttt{Information}, \texttt{Material}\}$ we may also have only $\{\texttt{Fault}, \texttt{Functioning}\}$, and introduce crossover relations like

$$\rho : \mathrm{T}_{\Sigma,\texttt{Fault}}(X_{\mathrm{u}})_{\mathrm{u}\in S} \times \mathrm{T}_{\Sigma,\texttt{Functioning}}(X_{\mathrm{u}})_{\mathrm{u}\in S} \to Q$$

with Q as a suitable quantale.

The set of types can be ordered, as also indicated in (Pimmler, 1994), namely, that $\texttt{Spatial}$ may have a special role in many applications.

Note how the many-typedness in (Eppinger, Whitney, Smith and Gebala, 1994) is more like a relation

$$\rho : E \times E \to 2_{Input} \times 2_{Feedback} \times 2_{Control} \times 2_{Addition}$$

and indeed not intended to be seen as a many-valued relation

$$\rho : E \times E \to \{Input, Feedback, Control, Addition\}$$

Many-valued extensions of the kind of enrichment appearing in (Pimmler, 1994) model now has bearing on “clustering elements into chunks” as outlined, but not detailed, in (Pimmler, 1994). These clusters are indeed called “chunks” in (Pimmler, 1994), where (Eppinger, Whitney, Smith and Gebala, 1994) prefers to call them “blocks”. These clusters are subsets $A \subseteq X$ together with their relations $\rho_A : A \times A \to 2$ restricted from the relation $\rho : X \times X \to 2$ so that $\rho_A(a_1, a_2) = \rho(a_1, a_2)$.

The notion of “clustering” in (Pimmler, 1994) is not made precise, and the thesis basically states that there are several approaches to clustering interaction matrices. There is a reference to (Eppinger, Whitney, Smith and Gebala, 1994), which was listed as a ‘forthcoming paper’. In that (Eppinger, Whitney, Smith and Gebala, 1994) paper, clustering as resulting also in ordering elements in general, ordering elements in blocks, and ordering blocks, is decribed in more detail, with references back to (Steward, 1965) and (Steward, 1981), but not further back to precedence matrices studied by (Barankin, 1953), and thereafter by (Marimont, 1959) and (Harary, 1960). In these historical path involving precedence matrices, relations are two-valued, and investigated graph-theoretically rather than logically. In these precedence matrices, X remains as a set of points not unravelled with respect to their possible content or appearance as expressions or terms, and $\rho : X \times X \to 2$ remains as a two-valued relation. Clustering is outside the scope of this paper, but a natural next step of further investigation. We should note that there is a wide range of approaches to clustering. Generally speaking, elements in a cluster share common features, and those features can be represented in various ways, numerically as well as logically. Many-valuedness should also be considered as an extension of imposing logical features with binary truth only. Many-valued clustering algorithms can be discretized to apply for many-valued relations over L^4, or over a suitable quantale Q, given a homomorphism $h : L^4 \to Q$. Directed graph (digraph) based clustering further has to distinguish between edge and node based clustering, or a mixture of the two. Digraph based clustering e.g. as in (Harary, 1960) is indeed graph-theoretical rather than logical and espression based, thus paying less attention expression and terms.

Relations between elements may lead to clusters, and thereby the distinction between elements and conglomerates of elements become apparent and subject to consideration. Algebraic properties of elements and sets provide information structure for subsequent logical treatments. Topological nearness of elements and products, as well as people and teams, are suitably modelled involving topological notions like neighbourhood, entourage, proximity and nearness. Entourages in uniform spaces are intuitively viewed two-dimensional or "relational" neighbourhoods. Nearness (Herrlich, 1974) extends proximities, modelling proximity of sets rather than metrically of elements. Topology in this application context can be seen as the abstracted notion of geometry and metric spaces, analyzing proximity and nearness from the viewpoint closer to the notion of contact relations and in (Düntsch and Winter, 2004) and (Düntsch and Winter, 2005), and as in this application context briefly outlined in (Johansson, Eklund, Kortelainen and Winter, 2018) and (Eklund, Kortelainen and Winter, 2019).

Multiple DSM domains, or multidomain matrices (MDM), involving component (Co), people (Pe) and activity (Ac), as described e.g. in (Browning, 2001), (Maurer, 2007) and (Eppinger and Browning, 2012), require elements in respective element sets X_{Co}, X_{Pe} and X_{Ac} to be related. Within the domain of components, a components relation is of the form $R_{CoCo} \subseteq X_{Co} \times X_{Co}$, whereas a relation between components and activity is of the form $R_{CoPe} \subseteq X_{Co} \times X_{Ac}$. In the case of R_{CoPe}, equivalently written as $\rho_{CoAc} : X_{Co} \times X_{Ac} \to 2$, we may want to assess quality for each component and production step (action) as weighed against the added value and the risk of upsetting the customer. We might then introduce the legend

$$A = \{Quality, MaterialValue, ManufacturingValue, CustomerSatisfaction\}$$

each item in the legend having a many-valued attribution from the many-valued truth set $L = \{To, Se, Mo, Mi, No\}$, so that many-valued relation between components and activities appears like $\rho_{CoAc} : X_{Co} \times X_{Ac} \to L^A$. Relational composition then immediately comes into play. We might have domain relation R_{CoCo} and R_{PePe}, together with multidomain relations R_{CoAc} and R_{AcPe}, enabling to arrive at relational compositions like $R_{CoCo} \circ R_{CoAc} \circ R_{AcPe}$, which in a many-valued setting will involve mappings between many-valuedness domains. This many-valued relational algebra is outside the scope of this paper, but will be developed in application oriented settings in future papers.

DSM and many-valuedness in practice

There are indeed many cases where a DSM may be combined with many-valuedness in order to achieve logic decisions that takes regard of overall consequences. A typical example is when implementing contactless quality measurements in an automated production line. Every time a robot handles a production article it will be known from which perspective an item is seen from a fixed point in any given moment. This is due to the need for the robot to follow its programming. Even though the robot may handle different articles, each article will need a specific programming, it will still be possible to know what objects are observed and in which way. Should a camera or other contactless sensor be placed next to the robots working area it will be possible to make observations of each object it handles, either passively or actively. Passively means that observations are made during the robots normal handling or, in the active case, the robot could move the object near a camera and actively show it from certain angles making the observations more accurate. There are ways to make several kinds of quality related measurements using cameras or similar sensors. Surface roughness, angles, diameters of holes, silhouettes or colours are just some examples. When the technical equipment is in place there will still be some issues to deal with. One being how to define quality, one is how to quantify quality and one being how to assess the impact of the measurement regarding the amount of manufacturing value that has been put in to it. For each production item there must be a rule base that clearly defines the parameters it needs to fulfil in order to be a functional article. For some articles the parameters may be very strict and for others they may be quite loose. Also the parameters may have different tolerances for a single object depending on its purpose and the next step in the production. One example is that holes might have very strict tolerances since it might be the place for a ball bearing seating while pressed angles might have very wide tolerances since they might be there for welding the article in to place and be forced in to place using a fixture.

Considering the quantification of the quality of an object the scale could be different for different parts of a production line and also differ from object to object. For simple objects like punched spacers, with very little production value added, there might be just two options; approved or not approved and the consequences of discarding individual items will likely be infinitesimal. For a production item that have gone through further production steps, like punch, fold and press, there might be a need to further refine the scale; approved, not approved or correction/manual inspection needed. There could be a chance that the original fold occasionally gets out of shape during press due to tensions in the materials and that it is deemed defendable to manually adjust objects with this specific deviation. Then we do not have a binary value but a level of functionality; fully functional, functional with correction, not functional. The corrections does not necessarily have to be made before the next step in the production, sometimes corrections can be made at a later stage, this makes it necessary to have an infrastructure that can connect a specific quality evaluation to a specific production item. Depending on the amount of value added to a specific production item, the economic consequences of simply discarding an item will wary e.g. in quantification of value-adding by manufacturing metrology, as described in (Savio, 2012). This means that after a certain point it will not be financially sustainable to just use a scale of faulty or functioning. Something more accurate is needed. The level of functionality used previously in the article is a good analogy but might be renamed into the level of correction. A production item with much added value might in the later stages of production have accumulated several points where correction is needed in order to achieve the level of quality that the customer demands. The total value of the item will decide the amount of correction that may be put in to it without the manufacturer risking its profit. The lost profit must also be put in relevance with the consequences of a missed shipment. In some cases it might be worth losing money on a single item in order to fulfil a larger customer contract. In this article we argue that a DSM could be a useful tool to keep control of all of these dependencies. Each product that a manufacturer makes will need its own DSM in order to optimise the actions after each quality measure.

Acknowledgement

This work is carried out within the projects *Production Centred Maintenance (PCM) for real time predictive maintenance decision support to maximise production efficiency*, funded by the Swedish Knowledge Foundation (Stiftelsen för kunskaps- och kompetensutveckling), *A digital twin to support sustainable and available production as a service (DT-SAPS)*, funded by Produktion2030, the Strategic innovation programme for sustainable production in Sweden, and *NORDIC Icing Center of Expertise (NoICE)*, funded by the the Interreg Botnia-Atlantica 2014-2020 programme. We gratefully acknowledge the support and funding.

References

Barankin, E.W., 1953. Precedence matrices, University of California Management Sciences Research Project, Research Report No. 26.

Browning, T.R., 2001. Applying the design structure matrix to system decomposition and integration problems: A review and new directions, IEEE Trans. Engneering Management 48 No. 3, 292-306.

Düntsch, I., Winter, M., 2004. Algebraization and representation of mereotopological structures, JoRMiCS 1, pp. 161-180.

Düntsch, I., Winter, M., 2005. A representation theorem for boolean contact algebras, Theoretical Computer Science 347(3), pp. 498-512.

Eklund, P., Gähler, W., 1992, Fuzzy Filter Functors and Convergence, Applications of category theory to fuzzy subsets (ed. S. E. Rodabaugh, E. P. Klement, U. Höhle), Theory and Decision Library B, Kluwer, 109-136.

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.

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.

Eklund, P. Kortelainen, J., Winter, M., 2019. Contact and influence between objects in a system-of-systems (extended abstract), LINZ2019, 38th Linz Seminar on Fuzzy Set Theory, Linz, Austria, February 5-8.

Eklund, P., Löfstrand, M., 2016. Many-valued logic in manufacturing, Position Papers of the 2016 Federated Conference on Computer Science and Information Systems (FedCSIS), ACSIS 9 (2016), 11–17.

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

Eppinger, S.D., Whitney, D.E., Smith, R.P., Gebala, D.A, 1994. A model-based method for organizing tasks in product development, Research in Engineering Design 6, 1-13.

Harary, F., 1960. On the consistency of precedence matrices, J. ACM 7 (3), 255-259.

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.
Johansson, M., Eklund, P., Kortelainen, J., Winter, M., 2018. The algebra, logic and topology of System-of-Systems, Proc. 20th International Dependency and Structure Modeling (DSM) Conference, Trieste (Italy), 15-17 October 2018, vol. 20, pp. 195–202.
Marimont, R.B., 1959. A new method of checking consistency of precedence matrices, J. Assoc. Comp. Mach. 6, 164-171.
Maurer, M.S., 2007. Structural Awareness in Complex Product Design, PhD thesis, Technischen Universität München, Munich, Germany
Savio, E., 2012. A methodology for the quantification of value-adding by manufacturing metrology, CIRP Annals 61 (1), 503-506.
Steward, D.V., 1965. Partitioning and tearing systems of equations, SIAM Numerical Anal., ser. B, vol. 2, no. 2, 345-365.
Steward, D.V., 1981. The design structure system: A method for managing the design of complex systems, IEEE Trans. Engineering Management, EM-28 no 3, 71-74.

Contact: P. Eklund, Umeå University, Department of Computing Science, 901 87 Umeå, Sweden, Phone +46 70 586 4414,
e-mail peklund@cs.umu,.se,
URL https://www.umu.se/en/research/groups/logic-and-applications/