THE MODULAR DESIGN EVALUATION MODEL:
SUPPORT FOR DECISION-MAKING IN CYBER-PHYSICAL SYSTEMS DESIGN

Marina Rantanen-Modéer

Saab Dynamics AB
SE-581 88 Linköping, SWEDEN
marina.rantanenmodeer@saabgroup.com

ABSTRACT

Complex Cyber-Physical Systems (CPS) design often starts from a set of legacy components and/or subsystems that help form the base of new systems. However, despite a finite set of possible combinations, the design space may be impractically large, which makes exhaustive analysis of all possible solutions for decision-making challenging. This paper proposes the Modular Design Evaluation Model (MDEM) as an approach to early design-stage system analysis, decision-making and implicit documentation through a proposed extension of the Design Structure Matrix (DSM). The MDEM integrates relative resource assessments of system elements for various design aspects of different type with an extension of the DSM that assumes a set of alternative CPS elements to be combined into new designs. The model helps reducing the design space and generates decision trees for alternative system evolution with an associated resource consumption, considering various design aspects and perspectives.

Keywords: Cyber-Physical Systems, Design Space Exploration, Systems Engineering Methodology

1 INTRODUCTION

The design of complex Cyber-Physical Systems (CPS) rarely starts from a completely blank page; usually there are legacy components or subsystems to consider for new configurations. CPSs integrate dynamical components with computational elements, which typically interact via intricate feedback loops, and may often form System-of-Systems (SoS), or so called Cyber-Physical Systems of Systems (CPSoS), as defined by Villar et al. (2020) among others. Existing system elements normally have corresponding legacy models on different level of detail and abstraction that have been built to answer specific design questions. Proposals for how to reuse such resources in the CPS design process are currently limited and often technology-specific, as suggested by Kim and Mosse (2008). Basing product evolution on existing design elements may be advantageous as it potentially reduces technical and economic risk and provides a backbone to relate new designs to in order to reduce the design space. However, the design space may despite a finite number of system elements be impractically large and design choices are therefore challenging to make given too many potential and viable configurations. Correspondingly, there is a challenge in knowledge transfer, i.e. dissemination of know-how about legacy subsystems and models that is often carried inside the head of the involved engineer. Unfortunately, documentation is - more often than not - meager, as suggested by Kotula (2000), or of too generic character, as suggested by Wulff et al. (2000). Consequently, it is challenging to retrace design decisions if the responsible engineer is no longer available to the team. To tackle these challenges, the contribution presented here proposes a methodology that:

1. reduces the design space for contract-based system design centered on combining alternative system elements,
2. supports decision-making given a set of quantifiable system performance targets,
3. implicitly stores design decisions in models that carry the information on which the decisions were made.

The paper is structured as follows: the first section provides insights on relevant aspects of model-based systems design and documentation. The subsequent section introduces the proposed methodology and describes it in depth using examples. The final section illustrates the proposal with a use case based on the analysis of a multi-agent system.

2 HIGH-LEVEL SYSTEMS DESIGN AND DOCUMENTATION

Model-Based Systems Engineering (MBSE) has become standard practice as system complexity has increased over time and prototyping at an early stage is either too costly or simply impossible, as stated by Wymore (2018). However, the notion of correct-by-design is a utopia as modelling is a flawed practice Sterman (2002). One of the many challenges of modeling is to capture *reality* and to synchronously combine the many formalisms and abstraction levels that are normally analyzed in CPS design (Modeer and Engell (2019)). This reality includes - among other things - a seemingly endless number of ways to achieve the same result. This section presents ideas from current literature on how to approach the vastness of CPS design through Design Space Exploration and some models for high-level system design and analysis.

2.1 Models for High-level Systems Design

The fact that CPSs rarely are designed from scratch implies that there are (plenty of) legacy systems with associated models to consider in the design process. A system designer is likely to have an interest in reusing such resources when investigating system effects of component changes. The reuse should preferably be possible to approach in a coarse manner avoiding more time-consuming, comprehensive methods. Therefore it is worth noting that high-level systems design may use seemingly banal models and tools. E.g. Schumann et al. (2010) discuss an approach to aerospace systems design using standard spreadsheets, i.e. tools that are common knowledge and require little extra effort for the various stakeholders of the system design chain to use, as opposed to specialized tools. However, experts will likely be drawn to using specialized tools to solve complex tasks, so standard office tools are to be regarded as an add-on for e.g. model-orchestration and a type of middle-ware for information propagation.

Another note should be made for lifetime extensions of existing systems, i.e. cases of evaluating system reuse versus partial upgrades, such as are common in e.g. the defence sector. Such cases imply complex information propagation chains to keep track of. Major system upgrades, or *overhaul* as defined by Barbosa and Souza (2018), require special attention to the system effects of these partial changes. Depending on the starting point, the choice of modeling approaches and methodologies will differ and often be a challenge to generalize.

2.2 Design Space Exploration

Design Space Exploration (DSE) refers to analyzing system alternatives before commencing the detailed design process as defined by Haveman and Bonnema (2013), and may for instance facilitate early-stage system alternative delimitation. However, the activity itself presents a challenge as the design space that must be explored typically is very large, as noted by Kang et al. (2010). A CPS may have an unreasonably (from a time and resource management perspective) large number of alternative designs that can fulfill the various requirements, which implies that enumeration of each design point is prohibitive. Literature suggests a number of frameworks and specialized tools for CPS DSE, e.g. a pareto optimal DSE framework presented by Amir and Givargis (2020) and "CyPhyML" presented by Neema et al. (2014).

2.3 Design Structure Matrix

The Design Structure Matrix - also referred to as the Dependency Structure Matrix (DSM) - has not only become a widely used modeling framework in academia in several areas of research, but has also gained traction in industry over the past couple of decades, as proposed by Browning (2015). It offers a comprehensive, yet compact representation of complex systems and is efficient for modeling and identifying dependencies between products, processes and organizations. The DSM was proposed by Steward (1981) for process design and analysis thereof, and has been updated, altered and extended on several occasions, as suggested by the survey presented by Browning (2015). The DSM is a square matrix, as illustrated in the example in Figure 1, whose columns and rows represent the parts or components of a system where interactions between any two elements are represented by some symbolism in the cells, as described by Eppinger and Browning (2012). Off-diagonal elements denote a direct relationship between the listed matrix elements.

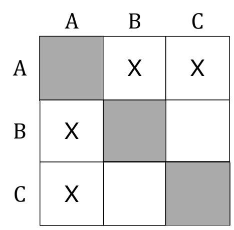

Figure 1: Off-diagonal matrix elements mark a direct relationship between the components A, B and C. Here A is related to both B and C, whereas B and C do not have any specific internal relationship.

2.4 Documentation in Systems Engineering

Continuous documentation of design-decisions has proven a necessity for efficient system design, and e.g. Tang et al. (2010) discuss the importance of support for architectural knowledge management activities, where documentation plays a major role. Analogously, Daun et al. (2014) make an argument for the necessity to document the context of knowledge, meaning that all documented information must be provided with context to be useful. Another crucial trait of effective documentation is easy-access to relevant information for all project members - only then will and can it be used properly, as concluded by Kinneging et al. (2020). Despite some efforts reported in current literature, effective implicit support mechanisms for documentation of decision-making in systems engineering are presently scarce. From empirical studies, one can observe resistance towards documentation in everyday engineering work as it is often perceived as a tedious task. In fact, Stettina and Heijstek (2011) describe the matter as "neglected and necessary", but conclude that when it comes to knowledge transfer, developers, despite having little interest in performing the documentation activities themselves, prefer access to comprehensive, written documentation of the system they are working on. Well-structured and proper descriptions of both technical and process perspectives may be ideal, but if the alternative is no documentation at all, middle-ground could be sought after. Current literature is lacking suggestions for algorithmic support for implicit documentation of e.g. design-decisions and modeling results and remains an open area of research.

3 THE MODULAR DESIGN EVALUATION MODEL - MDEM

This paper proposes the Modular Design Evaluation Model (MDEM) which constitutes a model-based methodology for high-level, early-stage design-evaluation for systems engineering design. The prerequisite for the MDEM is a set of well-defined system elements that may be integrated via a contract-based design approach (as described by e.g. Benvenuti et al. (2008)) with alternative parts to achieve various system improvements. In short, the proposal integrates relative resource assessments of design aspects of different type with an extension of the DSM that assumes a system of alternative elements. The DSM is chosen as it allows for coarse, yet effective assessments of all integral system elements in a contract-based fashion. From this, n-ary weighted decision trees are generated to represent the evolution of the various design alternatives. The way that the DSM is applied builds on the assessment of the compatibility of any two elements in the design set (i.e. no matrix element can be left unassessed) to enable an approach for quantification of technical feasibility of sub-system configuration. This provides - in addition to the relative figures for system evaluation - a visualization technique and graphical representation of the system for communication between internal groups (e.g. development team) and external groups (e.g. the customer).

The MDEM outputs two types of figures to support the evaluation of sub-system configurations: one for the total resource consumption based on technical feasibility and another for the weighted average of a chosen set of design aspects. The rationale of the MDEM is hence:

1. Compute the total resource consumption of and generate decision trees for the alternatives of the design evolution based on technical compatibility
   - i Generate the DSM, including the compatibility assessment of elements included in the analysis.
   - ii Generate the decision tree and sum up the resource consumption per branch. Optionally add "obsolescence trace".
2. Compute the weighted average of the total resource consumption considering multiple design perspectives
   - i Select aspects to be considered and quantify resource consumption of relevant activities. Exclude irrelevant (unviable) configurations.
   - ii Assign weights to all selected aspects.
   - iii Sum up all averages computed per relevant configuration.
3. Record decisions that were made based on the computations in step 1 and 2.

3.1 Definitions and Theorems

Definition 1. *A CPS consists of heterogeneous subsystems and may be partitioned into parts in a modular fashion with known interfaces such that alternative, redundant system elements can be considered in different configurations given a set of design constraints.*

Definition 2. *Resource consumption of any system element for any relevant design aspect may be defined as the "use of valuable or crucial development assets" and is assumed to have the potential to be quantified on a relative scale. Such resources may be combined for system-level analysis despite different type given that the scale remains the same throughout the assessment.*

Theorem 1. *Let $X = \{x_1, x_2, ..., x_n\}$ be a finite set of $n = 1, 2, ...$ CPS elements and $C = \{c_1, ..., c_{\frac{n^2-n}{2}}\}$ be a set of $\frac{n^2-n}{2}$ inter-compatibility assessments. Let $c = 0, 1, ..., j$ be a relative scale of choice to denote compatibility between any two elements in X, and let $c = -/\emptyset$ denote incompatibility. Let $X_{config} \subseteq X$ be a set that contains all necessary elements to form the sought after CPS. Then $X_{config} \forall C_{config} \neq \emptyset$ corresponds to a viable design solution.*

Theorem 2. *Let* $X_{config_\alpha} = (x_1 \vee x_{II} \vee ... \vee x_{III}) \wedge (x_{IV} \vee x_V \vee ... \vee x_{VI}) \wedge ... \wedge (x_{VII} \vee x_{VIII} \vee ... \vee x_{IX}) \forall x \in X$ *be the design rule by which the combination of elements in X is determined, as defined by Definition 1. Then the total number of configurations in X is given by Boolean algebra of the given formula for* X_{config}*. The subset of X in the first term of the design rule forms the root(s) of the decision tree(s) presented below.*

Theorem 3. *Let m be the number of elements in* X_{config} *and* $M = \frac{m^2-m}{2}$ *be the number of compatibility assessments associated with* X_{config}*. Then the resource consumption Res for* X_{config} *with an associated set* C_{config} *is given by* $Res = \frac{\sum_{j=1}^{m} c_j \in C_{config}}{M}$*, where j is the number of elements in* C_{config}*.*

Theorem 4. *Let* $A = \{A_1, ..., A_i\}$ *be a set of* $i = 1, 2, ...$ *design aspects considered for each design configuration* X_{config} *and let the associated resource consumption be* Res_{A_i}*, where each aspect holds a weight of* w_k *respectively. Then the total resource consumption* Res_{tot} *considering all chosen design aspects A is given by* $Res_{tot} = \sum_{k=1}^{i} Res_{A_k} * w_{A_k}$*, where* $Res_A = \frac{\sum_{j=1}^{m} c_j}{M}$*, and* $\sum_{k=1}^{i} w_k = 1$ *(all weights sum to one).*

3.2 Design Space Reduction Using an Extended DSM

CPS design normally assumes a set of alternative system components where *smart* choices must be made to achieve sought after system properties and meet any related performance targets. As an example, assume that a system is to be constructed from two subsystems with a couple of different alternative combinations, such that component X_1 must be combined with either component X_2 or X_3, but cannot be combined with both to form the final system. Formulated in accordance with Theorem 2 the design rule would then be $X_{config} = X_1 \wedge (X_2 \vee X_3)$, with a subsequent $1 * (1+1) = 2$ potential system configurations. Figure 2 illustrates how a DSM that includes complexity weights denoted for each matrix element relationship may be translated into propagation - or *decision* - trees for possible combinations (i.e. a display of the alternative design evolution with related resource consumption).

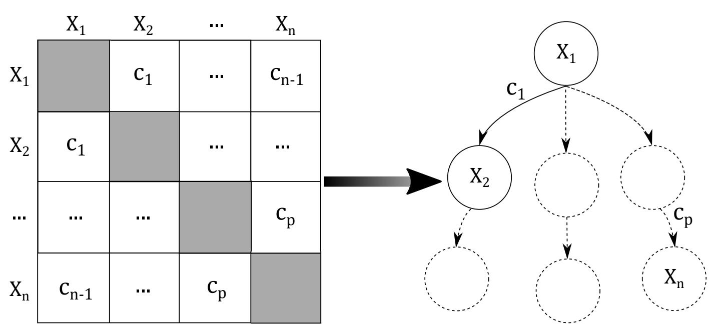

Figure 2: A finite set X of n partially redundant CPS elements in a DSM with associated relationships expressed as a set C of $p = \frac{n^2-n}{2}$ compatibility assessments. X may be represented as a decision tree, given a strict design rule that states restrictions on the combination of elements.

3.3 Strategy for Quantifying Properties of System Elements

The proposed MDEM is dependent on comprehensive compatibility assessments of system elements using a relative scale of estimated resource consumption C to combine all elements in question for any perspective of choice. The assessments must be made by domain experts and the numbers must be set in accordance with a mutual understanding of the contract between the system elements. As suggested by Theorem 1, the elements of C may be assigned any value $c = 0, 1, ..., \beta$ or be empty, as follows:

1	: low resource consumption - high compatibility
2	: higher resource consumption - lower compatibility
...	: ...
β	: maximum resource consumption - minimum compatibility
-/$\emptyset$	: no compatibility

Compatibility assessments are strictly made perspective by perspective, e.g. technical, operational, safety-related compatibility and so on. The proposed assessment figures are a measure of the relative effort to achieve satisfactory performance for the chosen perspective. There is a trade-off to be noted here: low resource consumption implies low extra developmental effort, which indicates considerable existing compatibility. Analogously, elements may be impossible or unreasonable to combine, in which case the relative compatibility is crossed out using a dash (-) or an empty set ($\emptyset$).

3.4 N-ary Weighted Decision Trees for Alternative Design Evolution

N-ary weighted, decision trees may be generated from the DSM to illustrate alternative system evolution and calculate the related resource requirements for each branch. Theorem 2 imposes a Boolean definition using the given set of system elements and suggests using the elements included in the first term of the rule as base for the tree(s). However, it is worth noting that there is freedom to choose the base element of any tree and that the design rule by which to assemble the system is not visible in the matrix. Furthermore, the tree may potentially have a large number of branches as it grows exponentially and depending on the design constraints, the tree may have a varying number of branches at each node.

For a more complex example, assume that a technical compatibility assessment is to be made for a system $X = \{A,B,C,D,E,F\}$. Theorem 2 gives that $X_{config_{EX}} = (A \vee B) \wedge (C \vee D) \wedge (E \vee F)$ and thereby $(1+1) * (1+1) * (1+1) = 8$ potential configurations. In other words, any viable configuration $X_{config_{EX}} \subset X$ must consist of either component A or B and C or D and E or F. The alternative system element sets $\{A,B\}$, $\{C,D\}$ and $\{E,F\}$ are assumed to have an associated contract, i.e. a clear definition of their expected behavior, as imposed by Definition 1.

When generating propagation trees for decision-making, the weights for the individual component compatibility assessment are added to each branch for each level of the tree. The weights for each component level in the tree is combined with the compatibility assessment of the component of the prior level, as illustrated in Figure 3. E.g. on the top level $B \rightarrow C = 2$, then on the second level $C \rightarrow F = 3$ and $B \rightarrow F = 1$. Some combinations may be unwanted or infeasible, as mentioned above, and any branch containing "-" will be crossed out, which is also illustrated in Figure 3.

3.5 Notes on Tackling Technical Obsolescence

Lifetime extensions of complex systems require attention to long-term effects of design choices. Components may become technically, legally or operationally obsolete within its expected lifespan, but may still out of various reasons be interesting to evaluate. The MDEM enables analysis of such obsolescence challenges by color-coding compatibility assessments for obsolescence directly in the DSM and the decision trees, which adds another dimension to the analysis. This in turn allows for analysis of the alternative to any potentially obsolete design evolution. I.e. if a branch is coded as obsolete, one may consider forking the design evolution at the first encounter of an obsolete element. This allows the designer to account for the resource consumption for investing in the (soon-to-be) obsolete design path and calculate the alternative cost.

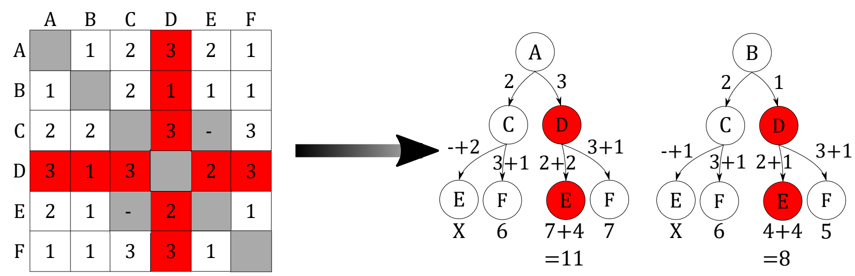

Figure 3: Two trees generated from a DSM. The weighted relationships between system elements add to the relative effort for possible combinations. An element that risks to become obsolete within the calculated life span of the CPS may be indicated in the decision tree by color. The total resource consumption for an obsolete branch adds the "next best" choice for the design evolution starting at the fork (i.e. at the first encounter of an obsolete element).

3.6 The Impact of Different Design Activities and Aspects

A key activity in early-stage systems design is setting appropriate performance targets as a step in the process to formulate high-level technical requirements. These targets may be concerned with e.g. the performance of the product itself, the development process and/or the organization. The various design aspects of interest often include, but are not limited to: cost (development, production, maintenance etc), time, risk as well as technical performance and product capabilities. The MDEM enables the designer to include any number of aspects in the evaluation. The two crucial activities here are to (1) make the correct assessment of the cost of the activity included in the design aspect and (2) to assign the appropriate weight to each aspect.

The assumption is that set of relevant aspects A (as defined by Theorem 4) may be identified and that each design activity may be quantified in relative terms. By quantifying cost, time, risk etc. on the same scale one enables the combination of aspects of otherwise completely different type. The MDEM imposes that each $x \in X$ is assessed for each aspect $a \in A$ separately, after which the total assessment for each configuration X_{config} (i.e. element combination) as an average for the aspect in question, can be made. The resource consumption for each X_{config_α} for each aspect is generated by assigning appropriate weights. Note that the *average* is calculated, rather than the sum, as the number of integral elements and activities may differ between aspects. Furthermore, it is worth noting that assigning the weights for each aspect is not trivial. This must be done in accordance with a comprehensive understanding for what the final system and its various stakeholders are aiming to achieve.

To illustrate this, consider the example in Figure 3 again. The relevant configurations are [A, C, F], [A, D, E], [A, D, F], [B, C, F], [B, D, E] and [B, D, F], as gathered from the two trees (note that two configurations are crossed out due to incompatibility). As input for decision-making, an analysis including cost, risk and time for each relevant configuration on the same scale could be made. In this example, assume that there are two separate activities in the development of each subsystem A through F for some aspect a_α, as indicated by in Table 1. The cost that each system element is associated with is assessed by relevant domain experts and then listed as illustrated in Table 1. The cost for the configurations is calculated by averaging the cost for each integral component, as presented on the right in Table 1. To retain the total resource consumption for the system design, each aspect is combined through weight coefficients for each configuration by applying Theorem 4 and as mentioned above, the weights play a key role as they have a significant impact on the resulting figure.

Table 1: Left: assessment and computation of the average resource consumption per system element. Right: computation of the average resource consumption per viable system configuration.

	Activity 1	Activity 2	Average
A	1	2	1.5
B	2	2	2
C	2	2	2
D	1	3	2
E	3	3	3
F	3	2	2.5

Relative resource consumption for components

Configuration	Average
A, C, F	$\frac{1.5+2+2.5}{3} = 2$
A, D, E	$\frac{1.5+2+3}{3} = 2.16$
A, D, F	$\frac{1.5+2+2.5}{3} = 2$
B, C, F	$\frac{2+2+2.5}{3} = 2.16$
B, D, E	$\frac{2+2+3}{3} = 2.33$
B, D, F	$\frac{2+2+2.5}{3} = 2.16$

Relative resource consumption for configurations

3.7 Implicit Documentation and Knowledge-transfer

Model-based documentation of design-decision is currently a fairly unexplored field of research. The MDEM aims at recording the reasoning for high-level design-decisions, which provides an implicit approach to documentation. To quantify the technical complexity of a combination of elements typically requires plenty of experience with and knowledge about the system in question. This implementation is not any different in that respect, which implies that the quality of the quantification is dependent on the knowledge that goes into the assessment and is done ad-hoc by the engineer in charge. However, the proposed model has the advantage that the visual aspect allows another approach to conveying information that facilitates knowledge transfer. The decision-making documentation consists of recording the rationale of any decisions through recording all high-level analyses made of the system. Using spreadsheets to build the MDEM, also provides any anslysis with a tool to save all system level reasoning in the same document. Any standard software tool providing spreadsheet properties enables this feature and can help link design choices to the reasoning based on the total configuration resource consumption calculation for all weighted design aspects. This implies that the documentation process lies in applying the model itself and by following the suggested steps the knowledge about the design is recorded in the process.

4 USE CASE: HIGH-LEVEL DESIGN OF A MULTI-ROBOT SYSTEM

This section provides an idea of the application of the MDEM. A standard spreadsheet tool is used to form the DSM, risk assessment, as well as calculate numerical values in each step of the analysis and store all generated information. The target is to evaluate a lab-scale system of mobile, deferentially steered Automated Guided Vehicles (AGV) that move between robotic production stations in a confined space, as described by Modeer et al. (2019). The AGVs carry vessels between the stations to pick up material to produce chemical products. The setup consists of three AGVs, four robotic production stations as well as various computers and components for communication. For the systems design, the following set of alternative subsystems are to be considered in a contract-based fashion:

- A positioning system based on either **(B)**: visual feedback via a camera OR **(C)**: radio-frequency identification (RFID)
- A control system based on either **(D)**: a decentralized architecture implemented on e.g. RaspberryPi OR **(E)**: a centralized design implemented on a PC
- A communication system based on either **(F)**: WiFi OR **(G)**: Bluetooth
- Off-the-shelf AGVs either from **(H)**: maker A that include WiFi capabilities OR **(I)**: maker B that include Bluetooth capabilities

	Positioning: Camera	Positioning: RFID	Architecture: decentralized	Architecture centrallized	Communication: WiFi	Communication: Bluetooth	AGV: maker A	AGV: maker B
Positioning: Camera		---	2	1	3	2	1	1
Positioning: RFID	---		1	1	1	3	1	1
Architecture: decentralized	2	1		---	1	1	1	3
Architecture: centrallized	1	1	---		2	2	2	1
Communication: WiFi	3	1	1	2		---	3	---
Communication: Bluetooth	2	3	1	2	---		---	1
AGV: maker A	1	1	1	2	1	---		---
AGV: maker B	1	1	3	1	---	1	---	

Figure 4: The elements in C_{config} for a multi-agent system analysis presented in the proposed application of the DSM. Dark gray matrix elements indicate the matrix diagonal. Light gray elements highlight incompatible combinations.

The rule for X_{config} is given as $X_{config} = (B \vee C) \wedge (D \vee E) \wedge (F \vee G) \wedge (H \vee I)$, with a subsequent $(1+1)*(1+1)*(1+1)*(1+1) = 16$ potential configurations. The analysis of the system design will be based on the technical compatibility of the different subsystems, where the resources to achieve satisfactory performance are assessed on a scale 1-3, where $c = 1$ is low resource consumption, $c = 2$ is intermediate consumption and $c = 3$ is high consumption, meaning that C_{config} is populated and then presented in the proposed application of the DSM. Furthermore, an assessment of the relative technical risk is to be made in addition to the technical compatibility, to guide the initial design choices. Using expertise in the combination of the various suggested technologies, a quantified relational DSM can be setup in accordance with Figure 4. Note that some elements are not compatible, which is expected as the AGVs cannot integrate with both suggested communication setups.

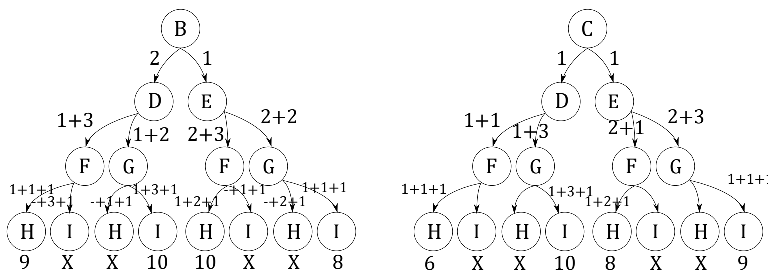

Figure 5: Left: Decision tree for the alternative evolution based on core component B. There are eight possible configurations, out of which four are unviable and the right-most has the lowest resource consumption. Right: Propagation tree for component C. A similar evolution can be observed as for component B.

From the DSM and the presented rule, one may now generate two decision trees, as presented in Figure 5. Component B and C are chosen as roots as they are core components here. The trees present the total resource consumption at the end of each branch and the paths that contain "-", i.e. an undesired subsystem combination, are crossed out. Respective tree illustrates the design evolution for each core component and one can conclude that there is a significant difference between the various alternatives. The least resource-consuming paths from each tree are: [B, E, G, I], which consists of the camera-based positioning system, centralized control architecture, Bluetooth-based communication and AGVs from maker B and sums up to 8 and [C, D, F, H], which consists of RFID positioning, decentralized control architecture, WiFi-based communication and AGVs from maker A and sums up to 6.

In this example, the system analysis is also to include the relative technical risk. This is first made on the basis of each subsystem for respective design activity, after which the risk is averaged for the chosen number of activities, as presented in Figure 6. Secondly the relative risk for each configuration is made based on the average computed for each subsystem in the previous step, by combining them and averaging again for each alternative configuration, as presented in Figure 6.

	A	B	C	D	E
A		Key features	Validation	Obsolescence	**Average**
B	Positioning: Camera	2	1	2	**1,67**
C	Positioning: RFID	3	3	3	**3,00**
D	Architecture: decentralized	2	2	0	**1,33**
E	Architecture: centrallized	1	2	0	**1,00**
F	Communi-cation: WiFi	1	2	2	**1,67**
G	Communi-cation: Bluetooth	2	3	2	**2,33**
H	AGV: maker A	1	2	1	**1,33**
I	AGV: maker B	1	2	2	**1,67**

	A	B	C	D
1		Average: B config		Average: C config
2	B, D, F, H	1,50	C, D, F, H	1,83
3	B, D, F, I	1,58	C, D, F, I	1,92
4	B, D, G, H	1,67	C, D, G, H	2,00
5	B, D, G, I	1,75	C, D, G, I	2,08
6	B, E, F, H	1,42	C, E, F, H	1,75
7	B, E, F, I	1,50	C, E, F, I	1,83
8	B, E, G, H	1,58	C, E, G, H	1,92
9	B, E, G, I	1,67	C, E, G, I	2,00

Figure 6: Left: Element risk assessment. Right: Configuration risk and viability assessment. The grayed out configurations are unviable as they include incompatible subsystems, as presented in Figure 5.

The total resource consumption for each configuration $Res_{tot,configx}$ is hence calculated separately as described by Theorem 4. Equation 1 illustrates how the total resource consumption for $X_{Bconfig1} = \{B, D, F, H\}$ is calculated, given equal weight ($w_{DSM} = w_{Risk} = 0.5$) of the two perspectives - technical compatibility and risk:

$$Res_{B,D,F,H} = \frac{D_{DSM} + F_{DSM} + H_{DSM}}{M_{DSM}} * w_{DSM} + \frac{B_{Risk} + D_{Risk} + F_{Risk} + H_{Risk}}{M_{Risk}} * w_{Risk} = \frac{(2+3+1)}{3} * 0.5 + \frac{(1.67+1.33+1.67+1.33)}{4} * 0.5 = 1.75 \quad (1)$$

The calculations, as presented by Equation 1, are repeated for all eight viable configurations and their results are presented in Table 2.

Table 2: Computation of average resource consumption for all viable configurations.

Configuration	Average
B, D, F, H	1.75
B, D, G, I	1.70
B, E, F, H	1.54
B, E, G, I	1.50
C, D, F, H	1.42
C, D, G, I	1.87
C, E, F, H	1.38
C, E, G, I	1.83

The computed values in Table 2 indicate that it is more resource effective to develop the second to last configuration, i.e. [C, E, F, H], which "costs" 1.38. However, the propagation trees provide a slightly different conclusion as [C, D, F, H] has a higher compatibility and sums up to 6, rather than [C, E, F, H], which sums up to 8, as illustrated in Figure 5. Which of the two configurations is then the better choice? Deeper knowledge about the system and its elements is needed. The MDEM outputs two figures for each configuration that provide guidance when making design decisions. However, the reasoning about the final choice requires more comprehensive understanding about the system and the environment that it acts in. The

MDEM helps to reduce the number of alternatives to consider. Technical compatibility and total resource consumption should be considered side by side for the final decision. By weighting the relative resource consumption and the relative risk using appropriate coefficients, the provided alternatives can be filtered based on those relative figures for $Res_{tot,system_X}$. Note that any number of aspect assessments can be added analogously with the risk assessment, i.e. for instance cost and time aspects by making separate analyses of these for each subsystem, before adding them to the total computation with appropriate weights. A DSM and related decision trees can be generated for other perspectives as well, such as operational, safety and security aspects, which may be of interest for various other project stakeholders. The spreadsheets that have been used to make the calculations also act as the basis of the documentation and separate tabs are kept for all design decisions that have been made based on the MDEM analyses. All formulated design decisions thereby directly link to the decision trees and computations - design decisions and their input are collected in the same place.

5 CONCLUSIONS

The paper proposes the Modular Design Evaluation Model (MDEM), a methodology that supports early-stage, high-level design of Cyper-Physical Systems (CPS). The methodology facilitates reduction of the often vast CPS design space, supports model-based decision-making given a set of quantifiable system performance targets, and includes an approach for implicit documentation of design decisions in high-level system models. The methodology is illustrated through conceptual system development of a multi-agent system. The main advantages of the proposed methodology are the visual and relative quantification properties of the system evaluation approach, that provides support for early-design-stage decision-making. The DSM is effective for the assessment of system element compatibility and is appropriate for design evaluation. However, the method is recommended for systems of a limited number of elements as it might not scale effortlessly to very large projects. Visualizing different design evolution alternatives trough N-ary weighted trees provides a tool for discussing design paths. A relative resource consumption figure is calculated for each design evolution, which enables a numerical comparison of viable design alternatives. In addition to design space reduction, the MDEM enables combination of different types of design perspectives to be considered in a unified analysis and automatically records crucial steps in the design process as a means for documentation. The MDEM hence provides an approach to evaluate system evolution such that resources may be focused on viable solutions.

REFERENCES

Amir, M., and T. Givargis. 2020. “Pareto Optimal Design Space Exploration of Cyber-Physical Systems”. *Internet of things* vol. 12, pp. 100308.

Barbosa, G. E. d. C., and G. F. M. d. Souza. 2018. “A Method for Defence Product Major Overhaul Analysis Focusing on Interfaces Redesign and Extended Life Supportability”. *International Journal of Production Research* vol. 56 (7), pp. 2523–2538.

Benvenuti, L., A. Ferrari, E. Mazzi, and A. Vincentelli. 2008. “Contract-based Design for Computation and Verification of a Closed-Loop Hybrid System”. In *International Workshop on Hybrid Systems: Computation and Control*, pp. 58–71. Springer.

Browning, T. R. 2015. “Design Structure Matrix Extensions and Innovations: a Survey and New Opportunities”. *IEEE Transactions on engineering management* vol. 63 (1), pp. 27–52.

Daun, M., J. Brings, B. Tenbergen, and T. Weyer. 2014. “On the Model-based Documentation of Knowledge Sources in the Engineering of Embedded Systems.”. In *Software Engineering (Workshops)*, pp. 67–76. Citeseer.

Eppinger, S. D., and T. R. Browning. 2012. *Design Structure Matrix Methods and Applications*. MIT press.

Haveman, S. P., and G. M. Bonnema. 2013. "Requirements for High Level Models Supporting Design Space Exploration in Model-Based Systems Engineering". *Procedia Computer Science* vol. 16, pp. 293–302.

Kang, E., E. Jackson, and W. Schulte. 2010. "An Approach for Effective Design Space Exploration". In *Monterey Workshop*, pp. 33–54. Springer.

Kim, J. E., and D. Mosse. 2008. "Generic Framework for Design, Modeling and Simulation of Cyber Physical Systems". *ACM Sigbed Review* vol. 5 (1), pp. 1–2.

Kinneging, T., R. de Graaf, S. Siebelink, and T. van Dijck. 2020. "The Documentation of Design Decisions in Engineering Projects: A Study in Infrastructure Development". *International journal of information systems and project management* vol. 8 (1), pp. 44–64.

Kotula, J. 2000. "Source Code Documentation: an Engineering Deliverable". In *Proceedings of the Technology of Object-Oriented Languages and Systems (TOOLS 34'00)*, pp. 505.

Modeer, M. R., and S. Engell. 2019. "Design and Validation of Cyber-Physical Systems through Model Abstraction". In *2019 International Symposium on Systems Engineering (ISSE)*, pp. 1–8. IEEE.

Modeer, M. R., S. Vette, and S. Engell. 2019. "Compensating Signal Loss in RFID-based Localization Systems". *IFAC-PapersOnLine* vol. 52 (8), pp. 142–147.

Neema, H., Z. Lattmann, P. Meijer, J. Klingler, S. Neema, T. Bapty, J. Sztipanovits, and G. Karsai. 2014. "Design Space Exploration and Manipulation for Cyber Physical Systems". In *IFIP First International Workshop on Design Space Exploration of Cyber-Physical Systems (IDEAL'2014), Springer-Verlag Berlin Heidelberg*, pp. 8.

Schumann, H., H. Wendel, A. Braukhane, A. Berres, A. Gerndt, and A. Schreiber. 2010. "Concurrent Systems Engineering in Aerospace: From Excel-based to Model-driven Design".

Sterman, J. D. 2002. "All Models Are Wrong: Reflections on Becoming a Systems Scientist". *System Dynamics Review: The Journal of the System Dynamics Society* vol. 18 (4), pp. 501–531.

Stettina, C. J., and W. Heijstek. 2011. "Necessary and Neglected? An Empirical Study of Internal Documentation in Agile Software Development Teams". In *Proceedings of the 29th ACM international conference on Design of communication*, pp. 159–166.

Steward, D. V. 1981. "The Design Structure System: A Method for Managing the Design of Complex Systems". *IEEE transactions on Engineering Management* (3), pp. 71–74.

Tang, A., P. Avgeriou, A. Jansen, R. Capilla, and M. A. Babar. 2010. "A Comparative Study of Architecture Knowledge Management Tools". *Journal of Systems and Software* vol. 83 (3), pp. 352–370.

Villar, E., J. Merino, H. Posadas, R. Henia, and L. Rioux. 2020. "Mega-Modeling of Complex, Distributed, Heterogeneous CPS Systems". *Microprocessors and microsystems* vol. 78, pp. 103244.

Wulff, I. A., B. Rasmussen, and R. H. Westgaard. 2000. "Documentation in Large-scale Engineering Design: Information Processing and Defensive Mechanisms to Generate Information Overload". *International Journal of Industrial Ergonomics* vol. 25 (3), pp. 295–310.

Wymore, A. W. 2018. *Model-based Systems Engineering*. CRC press.

AUTHOR BIOGRAPHY

MARINA M. A. RANTANEN MODÉER is a systems engineer and scientist at Saab AB, in Linköping, Sweden. She did her PhD research at the Technical University of Dortmund, Germany, on the modeling of Cyber-Physical Systems based on partial and heterogeneous models. Her research interests include Model Based Systems Engineering, Design Methodology and AI-driven Autonomy among other topics. Her email address is marina.rantanenmodeer@saabgroup.com.