Model-Based Diagnosis with FTTell: Assessing the Potential for Pediatric Failure to Thrive (FTT) During the Perinatal Stage

Natali Levi-Soskin[1*][0000-0002-2074-8214], Ron Shaoul[2,3], Hanan Kohen[1], Ahmad Jbara[1,4], and Dov Dori[1,5][0000-0002-2393-3124]

[1] Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Haifa, Israel
natali.levi@campus.technion.ac.il
hanank@technion.ac.il
ahmadjbara@gmail.com
dori@technion.ac.il
[2] Ruth Children's Hospital, Rambam, Health Care Campus, Haifa, Israel
[3] Faculty of Medicine, Technion, Israel Institute of Technology, Haifa, Israel
r_shaoul@rambam.health.gov.il
[4] School of Computer and Cyber Sciences, Augusta University, GA, USA
[5] Systems Design and Management, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract. Models have traditionally been mostly either prescriptive, expressing the function, structure and behavior of a system-to-be, or descriptive, specifying a system so it can be understood and analyzed. In this work, we offer a third kind—diagnostic models. We have built a model for assessing potential pediatric failure to thrive (FTT) during the perinatal stage. Although FTT is commonly found in young children and has been studied extensively, the exact etiology is often not clear. The ideal solution is for a pediatrician to input pertinent data and information in a single tool in order to obtain some assessment on the potential etiology. We present FTTell—an executable model-based medical knowledge aggregation and diagnosis tool, in which the qualitative considerations and quantitative parameters of the problem are modeled using a Methodical Approach to Executable Integrative Modeling (MAXIM)—an extended version of Object-Process Methodology (OPM) ISO 19450, focusing on the perinatal stage. The efficacy of the tool is demonstrated on three real-life cases, and the tool's diagnosis outcomes may be compared with and critiqued by a domain expert.

Keywords: Failure to Thrive (FTT) Potential · Executable Models · Systems and Software Engineering · OPM ISO 19450 · Model-Based Diagnosis

1. Introduction

Medical knowledge in general and pediatric medical knowledge in particular have been increasing significantly over the past decades. Therefore, it is hard or maybe even impossible for a pediatrician to be updated even in her or his field of expertise. In an attempt to provide a solution, some medical knowledge has been translated into computer interpretable formats as models or pseudo-code. Examples include GLIF [1], Arden [2], PROforma [3], EON [4], and GLARE [5]. The major problem with these frameworks is that they are not intuitive, and therefore medical doctors find them difficult to use. A 2016-published executable tool for representing medical knowledge and treatment protocols [6], which, to the best of our knowledge, is the state-of-the-art, translates medical data into Statechart models using the open-source tool Yakindu [7]. Since this tool is not executable and therefore does not provide for verification, an additional tool, Y2U [8], translates the Statecharts model from Yakindu to UPPAAL timed automata. In order to use this framework, medical doctors have to build the Statecharts model using Yakindu, then run Y2U to transform the model to UPPAAL timed automata, which can finally be executed, usually requiring help from a computer professional. Any change in the input data or the model structure, mandates updating the Statecharts model, re-running the Y2U tool, and executing the UPPAAL timed automata.

As the etiology of FTT is not well defined and has no consensus, models are subject to frequent changes. New factors might dynamically be added and existing ones might be changed or removed. Such modifications affect the model structure and will eventually invalidate the dynamic part due to lack of synchronization. The dynamic nature of FTT makes it difficult to develop a long-term write-once tool. A single tool that is based on a simple, coherent methodology, rather than a chain of disparate tools, will enable medical doctors to be actively involved in the tool development and its tuning as new factors and considerations emerge. Based on this motivation and our experience from previous works, including [9] and [10],we developed FTTell—a model-based improved, simple "one-stop-shop" medical diagnosis tool for assessing ("Telling") the potential for pediatric FTT during the perinatal stage. While different reasons may cause a child to deviate from the normal stature or weight for age and gender, it is not always the case that low weight or stature implies that a child fails to thrive [11], [12], as some of the reasons may be genetic- or nutrition-related. To diagnose FTT, the pediatrician should examine the child from many aspects, including parental (deprivation syndrome), prenatal growth, birth weight, and postnatal growth. Information about the parents, such as their heights, mother's nutrition status, and her emotional stress during the pregnancy, are also highly relevant. Our model represents succinctly and consistently the knowledge about the child at the perinatal stage—the period immediately before and after birth. It provides for collecting the required data, including the weight and length at different pregnancy stages, and getting an assessment of the child's FTT potential. Since diagnosis of FTT involves both qualitative and quantitative aspects, we have used MAXIM [10] , an extension of OPM. OPM [13], [14], is an ISO 19450 [15] model-based systems and software engineering language and methodology. Using MAXIM, we can represent qualitative and quantitative data in a model and execute it.

The main contribution of this research is the prototype model-based FTT diagnostic tool, which automates and expands FTT assessment. The tool uses data from an often-ignored and not thoroughly explored period—the perinatal stage, which is fed into a single executable model. Once any member of the medical staff has inserted the required data into the model, that member can easily execute the model in a single step, watch the results, track and analyze issues, and let the pediatric expert use it as a decision-support tool with no need for a computer professional's help. The model can be easily updated and synchronized with new relevant data and knowledge that is likely to emerge as the research on FTT is evolving.

The structure of the rest of this paper is as follows. In Section 2, we present MAXIM, which includes OPM and its computational extension, as well as the OPCloud OPM modeling environment. In Section 3, we present the model of the perinatal stage for diagnosing FTT. We present both the main diagram, in which the pediatrician can insert the required data and watch the results after model execution. Also presented are the refined diagrams, in which intermediate calculations are made and can be watched for deeper investigation. In Section 4, we summarize this research and suggest future research directions.

2. Method and tools

For developing FTTell we use MAXIM. MAXIM is based on OPM [13], [14] ISO 19450:2015 with extensions for computations. OPM, with its MAXIM extension, is a systems modeling approach [15] that represents the function, structure and behavior of any system using only two kinds of things (Figure 1): objects—things that exist, and processes—things that transform objects.

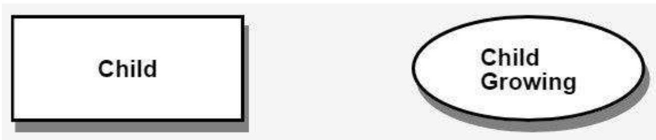

Figure 1: Example of an OPM object (left) and process (right)

Things can be physical (shaded) or informatical (not shaded). An informatical object can be valued, i.e., have a value, in which case it can also have units of measurement and an alias—a short, space-less name for use in formulae (see Figure 2).

Figure 2: An OPM physical object that represents a Child (left) and OPM informatical and computational object that represents a Weight of 15 kg and can be used in a formula shortly using its alias "w"

A process can be computational, in which case it accepts one or more object values as input parameters and performs a specified computation to produce a resulting object value. The difference between a physical process and an informatical and computational process is shown in Figure 3.

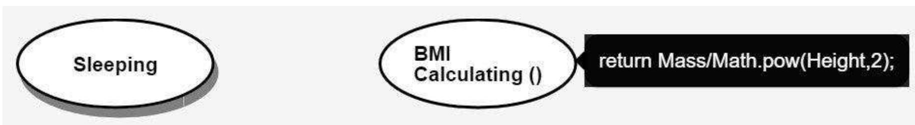

Figure 3: The OPM physical process **Sleeping** *(left) and the computational process* **BMI Calculating** *(right), which gets two valued objects,* **Mass** *and* **Height** *(not shown), and outputs a resulting value, the BMI object (not shown). The code of the computational process on the right appears in OPCloud while hovering over the process.*

OPM things are connected by links, which graphically express relations. Any OPM model consists of two parts: (1) the OPD set: a set of Object Process Diagrams (OPDs) and, (2) the OPL spec: a collection of sentences in a subset of English (or any other natural language), called Object Process Language (OPL). Each OPD construct—things connected by links—is reflected as one or more OPL sentences.

This bimodal representation caters to the dual channel assumption [16], [17]: the brain process information via both a visual and a verbal cognitive channels.

An OPM model can be presented at various levels of detail in different, interconnected views, each being an OPD. The top-level OPD is called System Diagram—SD, which usually consists of one systemic process and its operand – the object which that process transforms. Together, the process and the operand are the function of the system. Each OPD can be refined in one of several ways in order to expose deeper levels of detail. In this work, we use only process in-zooming to show subprocesses, of which the main process consists, and their temporal (sequential or parallel) execution order. MAXIM is developed and implemented as part of OPCloud [18], a collaborative cloud-based software environment for OPM-based modeling.

3. FTTell – Our OPM Model-Based FTT Diagnosis Tool

We start constructing our model by defining the main process, **Failure To Thrive (FTT) Diagnosing & Treating**, which is physical, as denoted in Figure 4 by the shading of the ellipse representing this process.

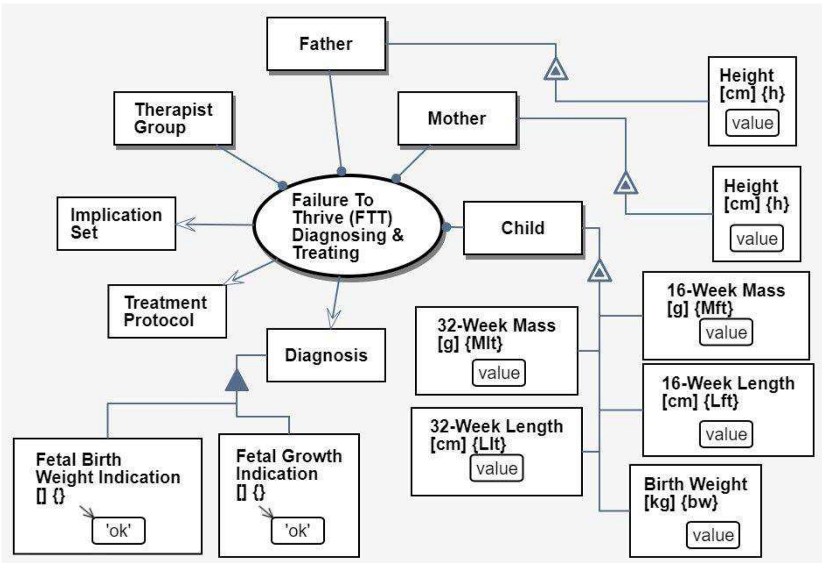

*Figure 4: SD of the FTTell system, showing the main process which is called **Failure To Thrive (FTT) Diagnosing & Treating**, the involved object set that serve as agents, and resulting objects*

The **Failure To Thrive (FTT) Diagnosing & Treating** process involves the objects **Child**, **Mother**, **Father**, and **Therapist Group.** Each one of the is a physical object, connected to the main process by an agent link—a line with a black lollipop on its process end. OPM agents are humans involved in the process. The **Failure To Thrive (FTT) Diagnosing & Treating** process outputs three results: **Diagnosis**, **Treatment Protocol**, and **Implication Set**. Each result is represented by an object connected to the main process by a result link, as shown in Figure 4.

The height of the child's parents can affect the FTT potential decision, so the **Mother** and **Father** have each a **Height** attribute. As Figure 4 shows, **Height** is a computational object with cm units and the alias h. The mother's and father's **Height** attribute objects are respectively connected to the **Mother** and **Father** by an exhibition-characterization link, serving as input values. The object **Child** has five perinatal-related input attributes: **Mass** [g] and **Length** [cm] at **16-weeks** and **32-weeks** pregnancy, as well as **Birth Weight** [kg].

The final result of the model execution is the informatical objects **Fetal Birth Weight Indication** and **Fetal Growth Indication**, which are parts of the **Diagnosis** informatical object. This is expressed in Figure 4 by aggregation-participation links—the lines with the black triangle in the middle. For the **Fetal Birth Weight Indication** object there are two possible result values: 'ok' and 'low'. For the **Fetal Growth Indication** object there are four possible values: 'ok', 'mother-dependent', 'mother- and child-dependent' and 'child dependent'. A result with value 'ok' means that there were no growth issues during the pregnancy. However, if the result is one of the other three options, then there were some problems, and an indication of possible reasons is then provided.

The small arrowhead pointing to each one of the values of **Fetal Birth Weight Indication** and **Fetal Growth Indication** symbolizes that these are the default values, which are 'ok'.

We refine the main process by zooming into it, exposing three subprocesses: **Diagnosing**, **Treatment Defining**, and **Implication Defining**. As in this work we focus on the **Diagnosing** process, we move on to Figure 5, in which we refine **Diagnosing** by zooming into it and exposing three new, lower-level subprocesses: **Perinatal Growth Examining**, **Postnatal Growth Examining**, and **FTT Diagnosing Determining**. For diagnosing FTT, both perinatal and postnatal data should be considered, but here we focus on the former, leaving the latter for future work. **Perinatal Growth Examining** uses data from the **Child** and provides two results: **Fetal**

Growth Indication and **Fetal Birth Weight Indication**. These objects have the default value 'ok', indicating that by default the growth and the birth weight of the fetus were normal.

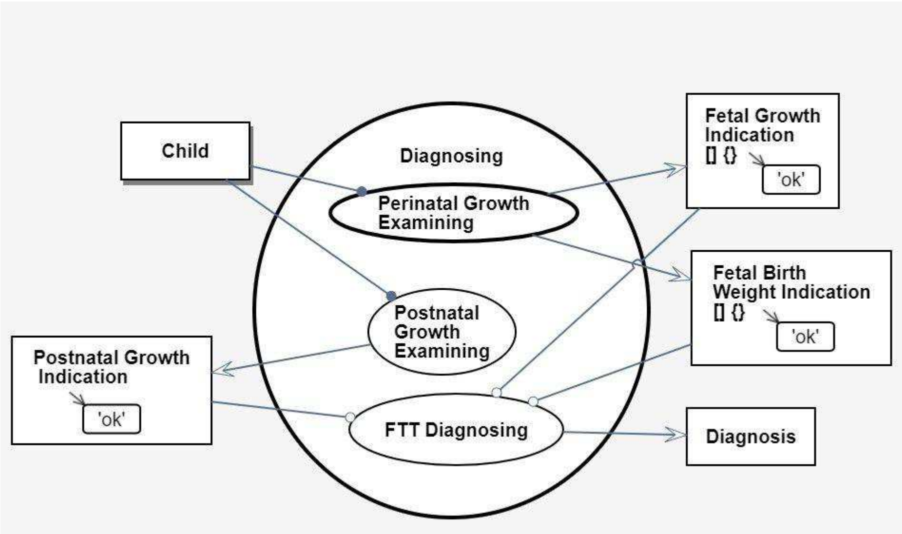

Figure 5: SD1.1- Diagnosing in-zoomed

Finally, in Figure 6, we zoom into the **Perinatal Growth Examining** process to determine the FTT potential. To this end, we calculate the ponderal index (PI) at 16 and 32 weeks, using the corresponding fetal **Mass** and **Length** at these time points while considering **Birth Weight** [11]. To perform these calculations, we wrote software code in OPCloud using the Typescript programming language.

The mathematical formula of PI is:

$$PI = \frac{mass\ (g)}{length\ (cm^3)} \times 100$$

The first subprocess, **First Trimester PI Calculating ()**, gets as input two values: **16-week Mass** {Mft} and **16-week Length** {Lft}. The result is written into the output object **First Trimester PI** {PIft}. The Typescript code is:

```
return (Mft/Math.pow(Lft, 3))*100;
```

Last Trimester PI Calculating () has a similar structure, using as inputs **32-week Mass** {Mlt} and **32-week Length** {Llt}, so the code for calculating **Last Trimester PI** {PIlt} is:

```
return (Mlt/Math.pow(Llt, 3))*100;
```

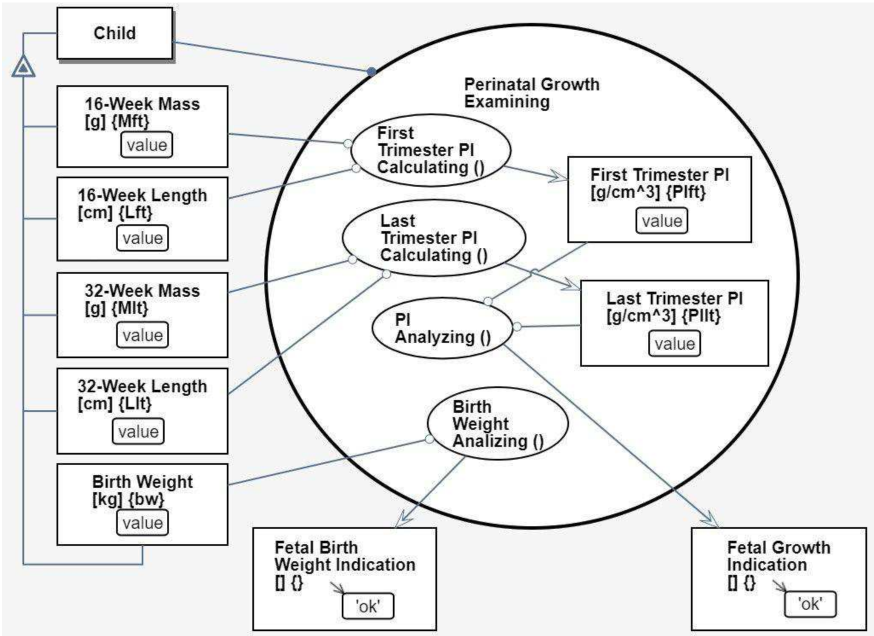

Figure 6: SD1.1.1 - Perinatal Growth Examining in-zoomed

To assess the FTT potential, **PI Analyzing ()** gets as input PIft and PIlt, and the result can be one of the following: (1) 'ok', which means no FTT, (2) 'mother-dependent', which can result from causes such as the mother's stress or bad nutrition during pregnancy, (3) 'mother- and child-dependent', i.e., FTT is caused by both the mother and some problem with fetal development, and (4) 'child-dependent', implying a fetal development problem, such as lack of proteins. The program code of this process is:

```
let ft = true, lt = true;
let result = 'ok';
if (PIft<2.32 || PIft>2.85) {
     ft = false;
}
if (PIlt<2.32 || PIlt>2.85) {
     lt = false;
}
if (!ft) {
     result = 'mother-dependent';
     if (!lt) {
```

```
     result = 'mother- and child-dependent';
     }
} else if (!lt) {
     result = 'child-dependent';
}
return result;
```

Finally, the **Birth Weight Analyzing ()** process uses the input **Birth Weight** {bw}, and the output is 'low' if the birth weight is less than 2.5 kg:

```
if (bw < 2.500) {
return 'low';
} else {
return 'ok';
}
```

Having developed and validated the FTTell model, we inserted into it the required data of each specific child, one at a time. The data is simulated, and was generated to test the efficacy of the FTTell system. Clicking on the execution button, we can follow tokens running visually through the different OPDs in a depth-first manner. Finally, the result values are updated in the model and can be exported to an Excel file.

Table 1 presents the results of particular data sets of three babies after executing the model. These are saved in an Excel file for documentation and can further serve for statistical analysis.

Table 1: Model execution results as saved in an Excel file

Fetal Birth Weight Indication	Fetal Growth Indication	Birth Weight	Last Trimester PI	First Trimester PI	32-Week Length	32-Week Mass	16-Week Length	16-Week Mass	
ok	ok	3.4	2.2	6.4	42.4	1702	11.6	100	Baby #1
low	ok	2.4	2.2	6.4	42.4	1702	11.6	100	Baby #2
low	mother- and child-dependent	2.4	2.3	2.9	41.3	1600	14	79.2	Baby #3

The first baby, Baby #1, was born with a weigh of 3.4 kg. It had a weight of 100 g and a length of 11.6 cm as a 16-weeks fetus, and a weight of 1702 g and a length of 42.4 cm as a 32-weeks fetus. This baby is defined as normal, with no FTT suspected.

Baby #2 has the same pregnancy values but a low birth weight of 2.4 kg, so it is classified by the system as 'ok' in the **Fetal Growth Indication** parameter, but as 'low' in the **Fetal Birth Weight Indication** parameter. Therefore, it may have some FTT suspicion and should be under follow-up. However, there is a chance that this baby's weight will catch up with the normal one in the next growth interval. Baby #3 has low length and mass values during the different perinatal stages, and is classified by the system with **Fetal Growth Indication** value 'mother- and child-dependent'. This result is determined by calculating PI values and analyzing them. Bad PI values at the 16-week pregnancy time point are usually due to mother-related reasons, such as bad nutrition and stress. Bad PI values at the 32-week pregnancy time point are caused usually by child-related reasons, which can be genetic causes, malnutrition, or lack of proteins. In addition, the third baby has a low birth weight. For all these reasons, it has high chances to suffer from FTT and has to be checked by an expert.

4. Conclusions and Future Work

Conceptual models have been used mostly for descriptive or prescriptive purposes, serving for understanding and specifying phenomena and systems, respectively. In this research, we propose using a diagnostic model – the application of conceptual modeling for the purpose of medical diagnosis. Currently, high-quality diagnosis of FTT potential depends on the experience and expertise of the pediatrician. Hence, as a proof-of-concept to our diagnostic modeling approach, we have developed, implemented, and started to test FTTell – a model-based diagnosis system for FTT potential assessment. To this end, we used MAXIM—an extended version of OPM ISO 19450, augmented with computational capabilities and implemented as part of OPCloud. Using this cloud-based modeling environment has enabled us to transition back-and-forth between qualitative and quantitative modeling in a seamless and effortless fashion. The underlying FTTell model embeds the necessary procedures and computations required for an objective decision-making process that a pediatric expert would perform. The input needed for the computation is easy to insert, and the execution is done by a click of a button in a friendly user interface.

Since OPCloud is a cloud-based tool and can be accessed from anywhere, the software is "zero client", making it executable from anywhere and at any time. Future research can take several directions: (1) Extend FTTell to include the postnatal stage, add input reasons that would yield mother-related FTT indications, and model an individually adapted treatment protocol that accounts for each child's data and indications. (2) Modify the model-based diagnostic approach to additional medical conditions and diseases that similarly require a stepwise procedure and computations using data supplied by the medical expert. (3) Apply a similar approach to the diagnosis of failure causes of technological products and systems, and how to repair or treat them.

Acknowledgments. The research in this paper was partially funded by the Gordon Center for Systems Engineering at the Technion, Israel Institute for Technology.

References

[1] R. A. Greenes, S. Tu, A. A. Boxwala, M. Peleg, and E. H. Shortliffe, "Toward a shared representation of clinical trial protocols: Application of the GLIF guideline modeling framework," in *Cancer Informatics*, Springer, 2002, pp. 212–228.

[2] T. A. Pryor and G. Hripcsak, "The arden syntax for medical logic modules," *Int. J. Clin. Monit. Comput.*, vol. 10, no. 4, pp. 215–224, 1993.

[3] J. Fox, N. Johns, and A. Rahmanzadeh, "Disseminating medical knowledge : the PRO forma approach."

[4] S. W. Tu and M. A. Musen, "Modeling data and knowledge in the EON guideline architecture Modeling Data and Knowledge in the EON Guideline Architecture," no. May 2014, 2001.

[5] P. Terenziani, S. Montani, A. Bottrighi, M. Torchio, G. Molino, and G. Correndo, "The GLARE approach to clinical guidelines: Main features," *Stud. Health Technol. Inform.*, vol. 101, no. May 2014, pp. 162–166, 2004.

[6] C. Guo, S. Ren, Y. Jiang, P. L. Wu, L. Sha, and R. B. Berlin, "Transforming Medical Best Practice Guidelines to Executable and Verifiable Statechart Models," *2016 ACM/IEEE 7th Int. Conf. Cyber-Physical Syst. ICCPS 2016 - Proc.*, 2016.

[7] "yakindu statechart tools." [Online]. Available:

https://www.itemis.com/en/yakindu/state-machine/.

[8] "Y2U," 2009. [Online]. Available: http://www.cs.iit.edu/~code/software/Y2U/.

[9] L. Li, N. Levi-Soskin, A. Jbara, M. Karpe, and D. Dori, "Model-Based Systems Engineering for Aircraft Design with Dynamic Landing Constraints Using Object-Process Methodology," *IEEE Access*, vol. 7, pp. 1–1, 2019.

[10] N. Levi-Soskin, S. Marwedel, A. Jbara, and D. Dori, "Towards Fusing Systems and Software Engineering : A Methodical Approach to Executable Integrative Modeling. submitted for publication."

[11] I. D. Schwartz, "Failure To Thrive: An Old Nemesis in the New Millennium," *Pediatr. Rev.*, vol. 21, no. 8, pp. 257–264, 2007.

[12] Z. Mei, L. M. Grummer-Strawn, D. Thompson, and W. H. Dietz, "Shifts in Percentiles of Growth During Early Childhood: Analysis of Longitudinal Data From the California Child Health and Development Study," *Pediatrics*, vol. 113, no. 6, pp. e617–e627, 2004.

[13] D. Dori, *Model-based systems engineering with OPM and SysML*. 2016.

[14] D. Dov, *Object-process methodology: a holistic systems paradigm*, 1st ed. Verlag Berlin Heidelberg: Springer, 2002.

[15] "ISO/PAS 19450 Automation systems and integration -- Object-Process Methodology," 2015. [Online]. Available: https://www.iso.org/standard/62274.html.

[16] R. E. Mayer and R. Moreno, "Nine ways to reduce cognitive load in multimedia learning," *Educ. Psychol.*, vol. 38, no. 1, pp. 43–52, 2003.

[17] R. E. Mayer, "The promise of multimedia learning: using the same instructional design methods across different media," *Learn. Instr.*, vol. 13, no. 2, pp. 125–139, 2003.

[18] "OPCloud." [Online]. Available: https://www.opcloud.tech/.