Physical-Informatical Essence-Duality-Aware Generic Modeling of Threat Handling Processes

Yaniv Mordecai
Technion – Israel Institute of Technology,
Haifa, Israel
yanivmor@technion.ac.il

Pathmeswaran Raju
KBE Lab, Birmingham City University,
Birmingham, UK
path.raju@bcu.ac.uk

Craig Chapman
KBE Lab, Birmingham City University,
Birmingham, UK
craig.chapman@bcu.ac.uk

Dov Dori
Technion, Israel Institute of Technology, Haifa, Israel
Massachusetts Institute of Technology, Cambridge MA, USA
dori@mit.edu

Abstract—**Systems interact with real world entities, and must hold internal representations of these entities in order to handle them appropriately. Physical-informatical essence duality (PIED) is the parallel existence of the entity as both the original, usually physical source, and its informatical representation, as held by each agent interacting with the entity. The distinction between the original external entity and its representation is critical for correct modeling and realization of complex interactions of cyber-physical systems with the real world. The implications of this distinction must be recognized and accounted for. Conceptual modeling semantics for the PIED problem make this distinction possible, structured, and well-defined in the system model. We review a formalism based on Epistemic Logic semantics, and a model-based framework based on Object Process Methodology, and demonstrate the applicability of our framework for generic modeling of threat handling processes, common to various cyber-physical systems and various types of threats, such as safety hazards, terror attacks, and cyber-attacks.**

Keywords- Model-based Systems Engineering; Physical-Informatical Essence Duality; Agent-Oriented Architecture; Object-Process Methodology; Epistemic Logic; Knowledge Representation; Cyber-Physical Systems; Threat Management;

I. INTRODUCTION

Systems are becoming ever more complex, multidisciplinary, and interconnected. Interactions with physical, biological, and social environments are constantly evolving, with a giant leap over the last two decades, with the emergence of cyber-physical systems (CPS). CPSs comprise virtual and physical segments of various controllers and actuators interacting with each other to achieve virtual and physical goals alike [1]. One of the fundamental principles of Cybernetics is the duality of the *physical world* and the *cyber world*, both of which are constituents of the *natural world* [2]. System architects increasingly follow robust and resilient design paradigms for systems that adapt or are easily adjustable to a variety of environments and working conditions. System integration, interconnectivity, and interoperability are critical for successful execution of complex cross-range processes [3].

Consequently, systems and agents interacting with their external environment and among themselves maintain a model of the environment. This model facilitates and affects the agent's capability to interact with the environment. Several agents interacting with the same environment can each hold a different model of this environment. Agents' representation of the external world can have varying levels of fidelity, related to the level of detail the agent is required to process, or capable of processing. The role of Informatical manifestations of "real-world" entities in natural, societal, and technical processes is one of the fundamental challenges in contemporary Cybernetics and Informatics [4].

Due to the distributed, stochastic nature of complex cyber-physical system-environment constructs, environmental entities have a dual nature. The entity is defined once by the original (most often physical) embodiment, and a second time by the representation thereof, as held by each relevant agent. The way an entity is captured, understood, or perceived by each agent in the system is defined by Mizzaro as "epistemic information" [5], i.e., information derived from knowledge and understanding. We define the dual existence of the original, physical entity and its informatical representation as the physical-informatical essence duality (PIED) [6]. PIED exists in virtually any type of system, including natural and societal contexts, complex systems, and systems-of-systems. This duality is also noted in the literature on Epistemic Logic [7] and Knowledge Representation [8]. Bar-Hillel's Semantic Information Theory has distinguished the meaning of content payload from the meaning of the physical carriers, namely, of information from data [9]. Hayles [10] provides an in-depth discussion on the implications of information-matter duality and dualism, including the abstraction of information from matter, the separation of information from the material source it reflects, and the separation of content from medium.

Conceptual modeling has significant importance in conveying and clarifying system structure, behavior, and functionality [11]. However, common modeling languages like UML and SysML [12], as well as available architectures adapted for cyber-physical systems [1], lack the semantics to capture this intricate yet critical notion, among other notions related to cyber-physical constructs [13]. Traditional system modeling approaches often fail to provide the semantics needed to capture and formally represent PIED.

Modeling languages grounded in the information systems domain mostly refer to entities as either physical or informatical, but not both. Typical UML models capture *actors* as top-level external entities, interacting with the system at the *use-case* level [12]. However, the actor is not associated with an internal representative *class*, actor designation is not replicated for subsystems at lower levels, and subsystems are rarely defined as actors of other subsystems. The simple notion that subsystems in a system constitute mutual actors is critical to the understanding of cross-structural processes in CPS, but PIED is not supported by conventional UML techniques.

This gap often leads to erroneous modeling, realization, and system behavior. In data integration, multiple representations must be accounted for [14], [15], but the interaction with the environment is not addressed. Cyber-physical constructs also consist of the actual physical entity, to which information holding components refer, and it must in general also be acknowledged and modeled. This complicates the problem of information integration due to the challenge of aligning information with 'real world' entities. System safety studies have shown that systemic misconceptions led to dire consequences and catastrophes [16].

Several studies related to PIED of model entities are concerned with precise detection of real-world events and information, especially in the fields of system safety [17], cyber security [18], and counter-terrorism [19], [20]. These are typical manifestations of the PIED problem, and all have in common the centrality of the threat identification and handling process.

In this paper we focus on the application of conceptual modeling semantics to reflect PIED in model-based cyber-physical systems engineering, specification, design, and development. The modeling semantics are provided in more detail in [6]. We define and analyze the implications of PIED on system models and review the formalism and model-based framework for defining, capturing, and representing the compound essence of entities. Our formalism consists of epistemic logic semantics, and is also object-oriented. Our conceptual modeling approach is based on Object-Process Methodology (OPM) [21], a holistic framework for complex systems modeling, and an emerging ISO 19450 standard.

The rest of this paper is organized as follows: Section II provides a brief description of OPM and its relevance to this study. Section III discusses the concept of PIED, its formal definitions and conceptual modeling semantics. Section IV illustrates PIED modeling in generic threat handling processes. Section V summarizes this paper and discusses future extensions.

II. OBJECT-PROCESS METHODOLOGY

Object-Process Methodology (OPM) [21] is a holistic framework for complex systems modeling, specification, design, and verification. OPM captures the functional, structural, and procedural aspects of the system in a single unified, dual graphical-textual view, which consists of a set of hierarchically organized and interconnected Object-Process Diagrams (OPD). Each OPD extends the OPD above it and adds more details, providing for inherent complexity management and reduction. Each OPD is accompanied by a textual computer-generated description in Object-Process Language (OPL) – a subset of English.

The primary building blocks in OPM are ***Objects*** and ***Processes***, collectively called ***things***. Objects are things that *exist* and can have ***states***. Processes are things that *occur* and transform objects: they generate and consume objects, or change their state. Objects are represented in OPDs as rectangles, while processes are represented by ellipses. States are represented as rounded-corner rectangles (“rountangles”) within their owning objects (Figure 1). OPM also provides a host of structural and functional links that capture the various object-object, object-process, process-object, and process-process relationships, presented in Figure 2

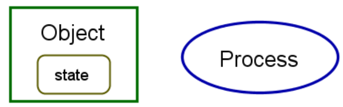


Figure 1OPM Basic Concepts: Object, Process, and State

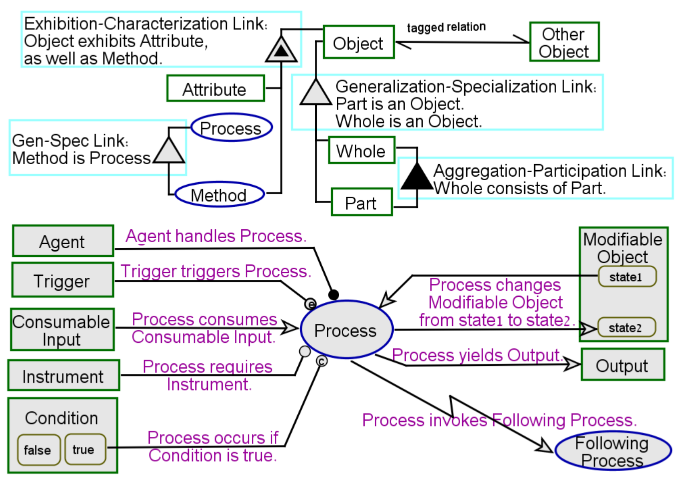


Figure 2OPM Structural and Procedural Relations

OPM’s freely available CASE tool, OPCAT [1] [22] automatically generates OPL sentences in response to visual model edits. The textual description is equivalent to the graphical view, allowing for enhanced model understanding. Any model fact that appears at least once in any diagram of the model is valid for the entire model. OPCAT also provides a built-in simulation engine which supports model validation, verification, and testing, and is especially useful in visualizing the execution of complex interactions.

OPM’s unified static-dynamic view has proven complexity management and alleviation capabilities that strongly support modeling of the PIED [23]. Inherent complexity management via hierarchical decomposition enables concealing or revealing representations, as required

[1] Downloadable for free from http://esml.iem.technion.ac.il/

for model understanding and communicating. The semantically equivalent graphical and textual views make OPM appealing to both sides of the human brain, catering to systems architects, domain experts, professionals and practitioners. The capability to generate meta-models—generic, multi-purpose models and patterns that can be instantiated and adapted for specific systems and problems—is a threshold condition for utilizing a language for metamodeling as practiced in this research. OPM's standardization as ISO 19450, as a basis for system and process modeling in ISO enterprise standards, enables accelerated dissemination of OPM for enterprise modeling. OPCAT fully supports the concepts and distinctions we explore and demonstrate, and allows for rapid and clear demonstration and visualization of PIED situations. PIED-aware models recently submitted by undergraduate students validate OPM's applicability for PIED.

III. Physical-Informatical Essence Duality (PIED)

In this section we review some basic formal definitions and understanding of the PIED. We define the formalism for describing this aspect, drawing on the semantics of epistemic logic, which are dedicated for capturing and describing knowledge and belief [7]. The notation is object-oriented and adjusted for conceptual modeling of complex engineering systems. We also briefly present the OPM-based conceptual modeling framework for expressing and handling PIED as part of system design and specification, previously described in [6].

There are two characteristics of the entity that are relevant in the PIED context: *affiliation* and *essence*. *Affiliation* denotes the entity as either: original and environmental (i.e., pertaining to the environment), or representational and systemic (i.e., pertaining to the system). *Essence* denotes the entity as either physical or informatical. While original entities are mostly physical, and representational ones are consequently informatical, all combinations of *affiliation* and *essence* are applicable and valid, especially in cyber-physical systems.

Due to the semantic importance of the essence and affiliation attributes, OPM supports their inherent designation and visualization. OPDs utilize shading to distinguish physical entities from informatical ones. Dashed contours mark environmental entities, while smooth contours mark systemic entities. These notations are visualized in Figure 3

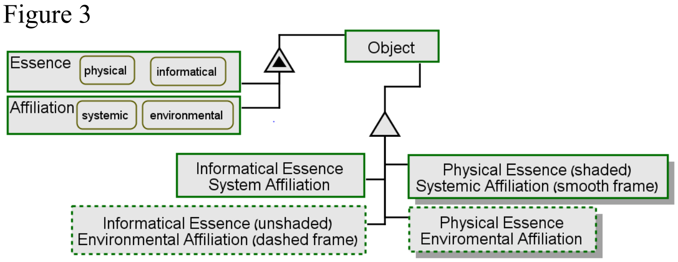


Figure 3OPM Distniction of Affiliation (contour) and Essence (shade)

In order to correctly approach PIED, we have to define the knowledge base of the system and each agent therein. Knowledge about the environment is required so that system agents can make decisions and perform actions. The knowledge base affects the outcome of reaction, action, and interaction. It includes what the agent needs to know, thinks, knows, and thinks it knows about each external entity.

Let E constitute an original entity. A representation of the entity, $R(E)$ consists of two aspects: (1) recognition of existence X, and (2) perceived state S, as defined in (1).

$$R(E) \equiv \langle X,S \rangle \tag{1}$$

Recognition of existence is dichotomous, so X can assume only 0 or 1. The perceived state S can assume a wide, combinatorial range of values and substates of the entity's characteristics, which can be recursively referred to via (1).

Let $A.R(E)$ denote the representation of E held by agent A. $A.R(E)$ can be defined only if E is found in A's knowledge base: $A.K_X(E)$ i.e., E is known to A when it exists. The recognition of existence is denoted by x. $A.R(E)$ can assume a state s only if s is defined in A's knowledge base.

$A.K_S(E)$ is the set of E's states that are known to A. It depends on A's recognition of existence, $A.K_X(E)$, but it is also distinguished from it, since agents may know the entity, but not necessarily all or even any of its state. These limitations are expressed in (2).

$$A.R(E) = \langle x | A.K_X(E), s | s \in A.K_S(E) \rangle \tag{2}$$

Each characteristic (i) of the entity E, $E.C_i$, constitutes a bona fide entity. The agent has to be aware of its existence, indicated by $K_X(E.C_i)$, and of its possible values, denoted by v, as listed in $K_S(E.C_i)$. This is shown in (3).

$$A.R(E.C_i) = \langle x | A.K_X(E.C_i), v | v \in A.K_S(E.C_i) \rangle \tag{3}$$

Representations of the same entity may often be integrated, synchronized, or compared, in order to improve their fidelity. A joint representation of two or more representations $n = 1,.., N$ is defined in (4) as $R|\{R_1(E),.., R_N(E)\}$

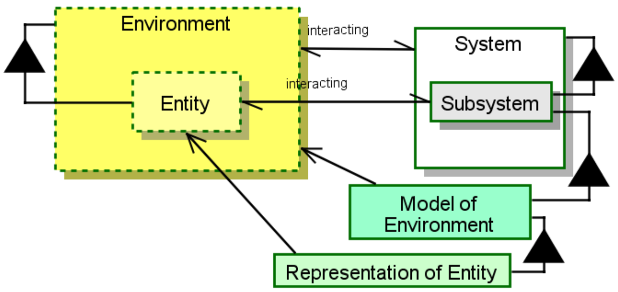
$$R \mid \{R_1,.., R_N\} = \langle h_X(\forall A_n.R(E).X), h_S(\forall A_n.R(E).S) \rangle \tag{4}$$

The agent's effect on the state of the original entity depends on its state as assumed by the agent. The entity's response to the effect can be coherent or incoherent, depending on the match between its actual state and the state perceived by the agent, as expressed in (5), where ξ denotes the coherence of the result of the entity's response function, $E.g(\,)$ to the agent's action function, $A.f(\,)$, which depends on the representation $R(E)$ and the state perceived by the agent $A.R(E).S$.

$$\xi\left[E.g\left(A.f\left(R\left(E\right)\right)\right)\right] = \begin{cases} 1, & A.R(E).S = E.S \\ 0, & A.R(E).S \neq E.S \end{cases} \tag{5}$$

The OPM metamodel for PIED-aware system models includes several hierarchical organized Object-Process Diagrams, followed by auto-generated OPL texts. Natural language names of model artifacts replace the mathematical symbols used above, making the text easier to read.

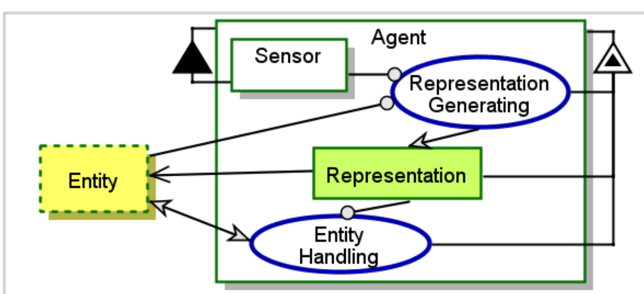
The topmost level depicted in Figure 4provides a high-level conceptualization of system-environment and subsystem-entity interaction, based on an internal model of the environment held by the system and its subsystems. This framework provides a starting point, an anchor and for additional modeling.



1. System and Subsystem are physical.
2. System consists of many Subsystems.
3. Environment and Entity are environmental and physical.
4. Environment consists of many Entities.
5. Environment and System are interacting.
6. Subsystem consists of Model of Environment.
7. Model of Environment relates to Environment.
8. Model of Environment consists of Representation of Entity.
9. Representation of Entity relates to Entity.
10. Many Entities and many Subsystems are interacting.

Figure 4. High-Level System-Environment Interaction

The next level of the model describes in Figure 5 the interaction of **Agent** with **Entity**. The agent needs the entity's **Representation** in order to handle the entity, using the **Entity Handling** process. The entity representation is generated by the **Representation Generating** process, and it requires the original **Entity**, as well as a **Sensor** to detect it.

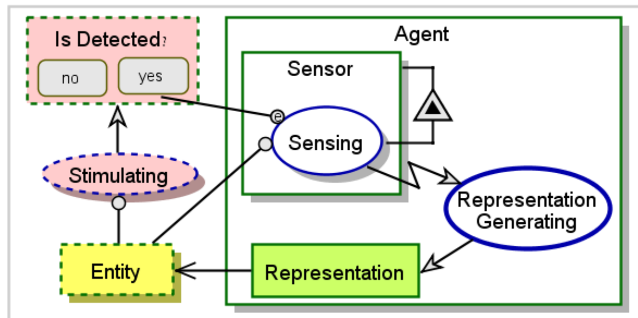


Representation Generating Representation
Representation Generating Sensor Entity
Representation Entity
Entity Handling Representation
Entity Handling Entity

Figure 5. OPD: Agent-Entity Interaction

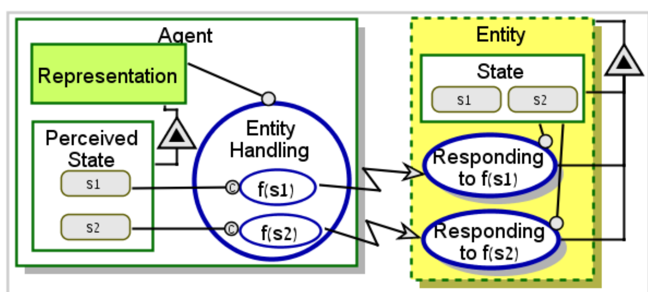
The **Representation Generating** process, elaborated in Figure 6 depends on the capability of the **Sensor** to detect **Entity** using its **Sensing** method. **Sensing** can occur only if stimulated. **Stimulating** is an external process which occurs randomly, reflecting the sensitivity of the **Sensor** to the **Entity** or change in an indicative attribute value it is designed to measure. The representation of **Entity** is generated in response to its successful detection.

The **Entity Handling** process, elaborated in Figure 7, depends on the existence of the representation and is governed by the **Perceived State** of the **Representation** of **Agent**. A dedicated handling method **f(s$_i$)** exists for each perceived state, Each method triggers the **Entity**'s respective **Responding to f(s$_i$)** process, which depends on the *original actual* **State s$_i$**.



1. Stimulating requires Entity.
2. Stimulating yields Is Detected?.
3. Sensing requires Entity and yes Is Detected?.
4. Sensing invokes Representation Generating.
5. Representation Generating yields Representation.

Figure 6. Object-Process Diagram: Representation Generation

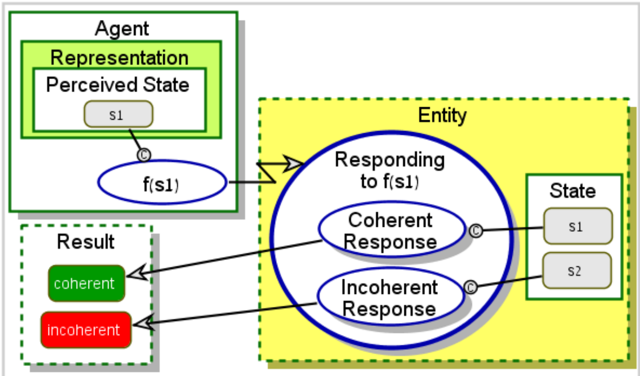


1. Entity exhibits State, as well as Responding to f(s$_i$).
2. Entity Handling consists of f(s1) and f(s2).
3. Entity Handling requires Representation.
4. f(s$_i$) occurs if Perceived State is s$_i$.
5. f(s$_i$) invokes Responding to f(s$_i$).

Figure 7. Object-Process Diagram: Entity Handling

The **Entity**'s **Responding to f(s$_i$)** process is elaborated in Figure 8. Coherence is obtained when the entity's **State** is equal to the agent's representation's **Perceived State**. Otherwise, the **Result** is incoherent. The result of the interaction can affect the representation of the original entity. Detecting the result (with appropriate means), representing it, feeding it back into the system, and updating the original representation, are parts of a dedicated process,

Representation Refining, which is not shown in this preliminary metamodel.



1. Result can be coherent or incoherent.
2. $f(s_i)$ invokes Responding to $f(s_i)$.
3. Responding to $f(s_i)$ consists of Coherent Response and Incoherent Response.
4. Coherent Response occurs if State is s_i.
5. Coherent Response yields coherent Result.
6. Incoherent Response occurs if State is s_j, $i \neq j$.
7. Incoherent Response yields incoherent Result.

Figure 8. OPD: Entity Responding to Agent Action

IV. PIED-Aware Modeling of Generic Threat Handling Processes

The formalism and OPM design pattern introduced in the previous section clearly facilitate capturing PIED. We now demonstrate the applicability of the metamodel we have presented to practical Cyber-Physical settings. The example we discuss concerns the generic process of threat management and control, relevant to many Cyber-Physical systems with these capabilities. A threat is an entity which is capable of harming system or stakeholder assets and activities.

A PIED-aware modeling approach includes the consideration of the real potentially threat-posing object (or simply **Threat**) as an external, separate entity, and the representation of the threat, **Threat Profile** as held by the agent **Threat Manager**, and generated by the **Threat Profile Generating** process, in response to **Threat Detecting.** This is illustrated in Figure 9.

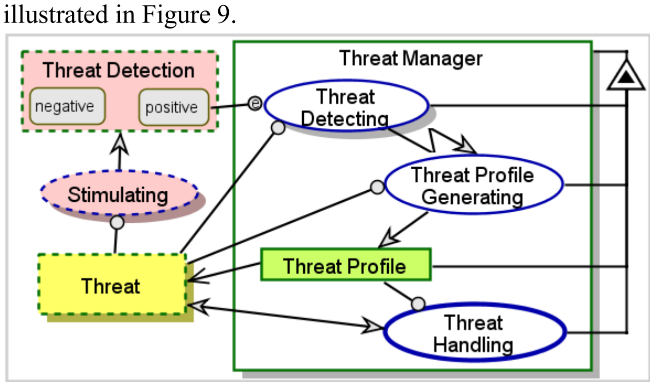


Figure 9. OPD: Threat Manager: Threat Detection and Representation

The **Threat Handling** process is illustrated in Figure 10. It includes **Attacking** the **Threat** if it is perceived as threatening, and **Ignoring** the **Threat** if it is perceived as unthreatening. Each action by the **Threat Manager** is responded to by the **Threat** with an adequate **Responding** process. The result of **Ignoring** the **Threat** is elaborated in Figure 11, where a coherent **Result** means **Causing No Harm**, while an incoherent **Result** means that the **Threat** is actually threatening, as opposed to what the **Threat Manager** thought, and therefore it is **Causing Harm**. Finally, a complete OPL description of the Threat Handling Meta-Model is provided in Figure 12. The model-level OPL text covers all modeling aspects visualized in the OPD diagrams of the model.

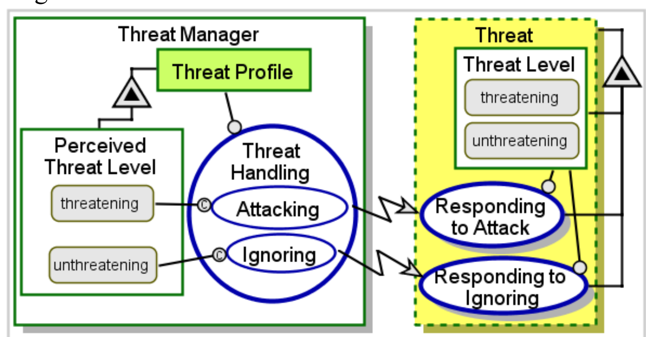


Figure 10. OPD: Threat Manager: Threat Handling

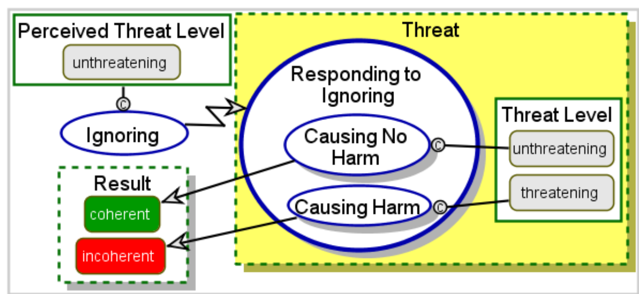


Figure 11. OPD: Threat: Responding to Threat Manager Ignoring

V. Summary

This paper introduces the Physical-Informatical Essence Duality (PIED) problem, formalism and conceptual modeling semantics to handle and contain it when it applies to various entities in systems and models. This problem is significant and must be handled appropriately especially in Cyber-Physical systems, in which systems constantly interact with the 'real world'. We have applied the modeling of PIED to the generic Threat Management process exhibited by many agents in various domains, e.g. safety, counter-terrorism, and cyber security. Future work will extend the PIED-aware model with layers reflecting aspects like feedback, conflict resolution, refinement and fusion of representations, and embedded coherence analysis. We also intend to further elaborate the Threat Management model and apply it as a metamodel to various threat handling situation in practical systems, especially in the fields of defense, health, and sustainability.

1. Threat is environmental and physical.
2. Threat Manager is physical.
3. Stimulating requires Entity.
4. Threat exhibits Threat Level, as well as Responding to Ignoring and Responding to Attack.
5. Threat Level can be threatening or unthreatening.
6. Responding to Ignoring is physical.
7. Responding to Ignoring consists of Causing No Harm and Causing Harm.
8. Responding to Ignoring requires Threat Level.
9. Responding to Ignoring zooms into Causing No Harm and Causing Harm.
10. Causing No Harm is physical.
11. Causing No Harm occurs if Threat Level is unthreatening.
12. Causing No Harm yields coherent Result.
13. Causing Harm is physical.
14. Causing Harm occurs if Threat Level is threatening.
15. Causing Harm yields incoherent Result.
16. Responding to Attack is physical.
17. Responding to Attack requires Threat Level.
18. Threat Manager exhibits Threat Profile, as well as Threat Handling, Threat Profile Generating, and Threat Detecting.
19. Threat Profile can be exists or nonexistent.
20. Threat Profile exhibits Perceived Threat Level.
21. Perceived Threat Level can be threatening or unthreatening.
22. Threat Profile relates to Threat.
23. Threat Handling consists of Ignoring and Attacking.
24. Threat Handling requires Threat Profile.
25. Threat Handling affects Threat.
26. Threat Handling zooms into Attacking and Ignoring.
27. Attacking occurs if Perceived Threat Level is threatening.
28. Attacking invokes Responding to Attack.
29. Ignoring occurs if Perceived Threat Level is in existent and Perceived Threat Level is unthreatening.
30. Ignoring invokes Responding to Ignoring.
31. Threat Profile Generating requires Threat.
32. Threat Profile Generating yields Threat Profile.
33. Threat Detecting is physical.
34. Threat Detecting requires Threat and positive Threat Detection.
35. Threat Detecting invokes Threat Profile Generating.
36. Result is environmental and physical.
37. Result can be coherent or incoherent.
38. Threat Detection is environmental.
39. Threat Detection can be positive or negative.
40. Threat Detection triggers Threat Detecting when positive.
41. Stimulating is environmental and physical.
42. Stimulating requires Threat.
43. Stimulating yields Threat Detection.

Figure 12. OPL Text for the Threat Handling Meta-Model

ACKNOWLEDGMENT

This research was partially funded by EU FP-7 VISIONAIR Project Grant 262044. The authors thank the anonymous referees for their useful advice.

REFERENCES

[1] Y. Tan, S. Goddard, and L. C. Pérez, "A prototype architecture for cyber-physical systems," *ACM SIGBED Rev.*, vol. 5, no. 1, pp. 1–2, Jan. 2008.

[2] Y. Wang, W. Kinsner, and D. Zhang, "Contemporary cybernetics and its facets of cognitive informatics and computational intelligence.," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics*, vol. 39, no. 4. pp. 823–33, Aug-2009.

[3] A. P. Sage and C. L. Lynch, "Systems integration and architecting: An overview of principles, practices, and perspectives," *Syst. Eng.*, vol. 1, no. 3, pp. 176–227, 1998.

[4] K. Kolin, "Philosophy of Information and the Fundamentals of Informatics," in *The Third International Conference "Problems of Cybernetics and Informatics,"* 2010, pp. 290–293.

[5] S. Mizzaro, "Towards a theory of epistemic information," *Inf. Model. Knowl. Bases*, 2001.

[6] Y. Mordecai, C. Chapman, and D. Dori, "Conceptual Modeling Semantics for the Physical-Informatical Essence Duality Problem," in *IEEE International Conference on Systems, Man, and Cybernetics - SMC2013*, 2013.

[7] J. Hintikka, "Individuals, possible worlds, and epistemic logic," *Nous*, vol. 1, no. 1, pp. 33–62, 1967.

[8] F. Van Harmelen, V. Lifschitz, and B. Porter, Eds., *Handbook of knowledge representation*. Elsevier, 2008.

[9] Y. Bar-Hillel and R. Carnap, "Semantic Information," *Br. J. Philos. Sci.*, vol. 4, no. 14, pp. pp. 147–157, 1953.

[10] K. Hayles, *How we became posthumans*. The University of Chicago Press, 1999.

[11] J. A. Estefan, "Survey of model-based systems engineering (MBSE) methodologies," *Incose MBSE Focus Gr.*, vol. 25, 2007.

[12] L. Favre, *UML and the Unified Process*. IRM Press, 2003.

[13] E. a. Lee, "Cyber Physical Systems: Design Challenges," *2008 11th IEEE Int. Symp. Object Component-Oriented Real-Time Distrib. Comput.*, pp. 363–369, May 2008.

[14] F. Saltor, M. Castellanos, and M. García-Solaco, "Suitability of datamodels as canonical models for federated databases," *ACM SIGMOD Rec.*, pp. 44–48, 1991.

[15] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati, "Information integration: Conceptual modeling and reasoning support," in *3rd IFCIS International Conference on Cooperative Information Systems*, 1998.

[16] N. Leveson, *Engineering a safer world*. MIT Press, 2011.

[17] C. Tomlin, S. Member, G. J. Pappas, and S. Sastry, "Conflict Resolution for Air Traffic Management□: A Study in Multiagent Hybrid Systems," vol. 43, no. 4, pp. 509–521, 2000.

[18] V. Jaiganesh, S. Mangayarkarasi, and P. Sumathi, "Intrusion Detection Systems□: A Survey and Analysis of Classification Techniques," vol. 2, no. 4, pp. 1629–1635, 2013.

[19] M. Chmielewski, A. Gałka, P. Jarema, K. Krasowski, and A. Kosiński, "Semantic Knowledge Representation in Terrorist Threat Analysis for Crisis Management Systems," pp. 460–471, 2009.

[20] G. J. Victor, M. S. Rao, and V. C. Venkaiah, "Intrusion Detection Systems - Analysis and Containment of False Positives Alerts," *Int. J. Comput. Appl.*, vol. 5, no. 8, pp. 27–33, 2010.

[21] D. Dori, *Object-Process Methodology: A Holistic Systems Approach*. Springer, 2002.

[22] D. Dori, C. Linchevski, R. Manor, and O. M. Opm, "OPCAT – An Object-Process CASE Tool for OPM-Based Conceptual Modelling," in *1st International Conference on Modelling and Management of Engineering Processes*, 2010, pp. 1–30.

[23] D. Dori, I. Reinhartz-berger, and A. Sturm, "Developing Complex Systems with Object-Process Methodology Using OPCAT 1 The Basis□: Object-Process Methodology," pp. 570–572, 2003.