Third International Engineering Systems Symposium
CESUN 2012, Delft University of Technology, 18-20 June 2012

Risk-Oriented Systems Engineering: Integrating Risk into Systems Modeling with OPM

Yaniv Mordecai[1] and Dov Dori[1]
[1]Technion – Israel Institute of Technology, Haifa 32000

yanivmor@tx.technion.ac.il , dori@ie.technion.ac.il

Abstract. *System Design and Risk Management are two critical Systems Engineering processes, that are currently insufficiently integrated. Risk analysts use separate tools, techniques and semantics, often leading to irrelevance of the risk management effort. We introduce a framework for the fusion of system design, system configuration and risk analysis, called ROSE – Risk Oriented Systems Engineering. ROSE incorporates risk identification, modeling, analysis, mitigation and control aspects into the traditional system design process. This way, risk management becomes synchronized with, and embedded in, system design, while system design is enriched by the perspectives of risk identification, handling and control. Subsequently, as the design model becomes a system deployment and configuration model, operational configuration management and risk management are conducted jointly. Our framework facilitates the communication and collaboration between risk analysts and system engineers, and increases the ability to identify, understand, assess and monitor risks as they emerge and evolve throughout the system's lifecycle. Our underlying modeling framework is Object-Process Methodology (OPM), a bimodal visual and textual structured conceptual modeling language, enhanced by risk modeling constructs. OPM, an emerging ISO standard for system modeling and design, features a free CASE tool, OPCAT, which allows for fast implementation in both the system engineering and risk management domains.*

Keywords. *Risk, Risk Management, Risk Modeling, Robust Systems Design, System Configuration, Object Process Methodology, Conceptual Modeling.*

1 Introduction

Robust systems design allows for reduction of development cycle cost and duration, enables gradual product adaptation, and improves response to predicted or evolving market requirements (Diaz, 1998). However, robustness often increases initial time-to-market and development costs, and balancing these objectives is a continuous product management decision making problem (Krishnan & Ulrich, 2001). Robust design leads to operational configurability and flexibility. Due to its robust design, Lockheed-Martin's C130 "Hercules" military carrier aircraft, for instance, whose name is synonymous with versatility, may be configured for various missions, including troop airlift, paradropping, medical evacuation, search and rescue, cargo, scientific and weather reconnaissance, aerial refueling, and aerial firefighting (Wikipedia – the free encyclopedia).

System configurability can be achieved in various ways, e.g., editable system parameters, support for activation and deactivation of features or components, modularity and the ability to reshape, reorder or reconstruct the system based on its building blocks, and even the ability to reprogram a system and modify or extend existing functionality. Various IT products, like knowledge management portals, messaging systems, IT resource monitoring & control, and security software, are provided to clients off-the-shelf, while the actual shaping of the solution is done almost completely through configuration, calibration and capability generation, through the product's inherent configurability, with little or no extra development.

Risk Management is a key success factor in both the Project Management and Systems Engineering domains (PMI, 2006; INCOSE, 2004). Risk Management aims at reducing the probability of occurrence of risky processes and their adverse impact on stakeholder objectives and assets. Project Risk Management – PRM (Chapman and Ward, 2003) focuses on reducing delays and cost overruns, while satisfying specification (spec) and quality requirements. Operational Risk Management – ORM (Hoffman, 2002; Haimes, 2009), is concerned with assuring such objectives as reliability, safety, security, availability, and business continuity (some of the so-called "ilities") in operational settings under risk. It is up to the operating organizations to prepare contingency plans and procedures to handle such risks, using guides and standards (NASA 1995; Stoneburner et al., 2004; ISO/IEC/IEEE, 2006).

Robust design is a useful risk management strategy with a double effect: it is useful in handling developmental risks like vague or contradicting requirements, and it equips the system with the necessary flexibility to face various operational challenges and working conditions. Nevertheless, flexibility also incurs risk, as it may result in increased development cost and duration, and in extreme configurations which are not appropriately risk-hedged. Providing full built-in risk response capability to every configurable state is difficult to impossible, when the state-space is combinatorial.

System risk analysis requires quantitative, probabilistic techniques (Cooke, 1991; Bedford & Cooke, 2001), and dedicated system-oriented methods (Haimes, 2008), in addition to classical risk analysis methods, such as Fault-Tree Analysis (FTA), Failure Mode Effect Critical Analysis (FMECA) (Haimes, 2009), and Hazard and Operability (HAZOP) (Redmill et al., 1999). Analytical risk-integrated system modeling attempts to define the system's (multi-)objective function, while capturing risk, with mathematical building blocks, like input, output, state variables, decision (control) variables, and random variables. System vulnerability is a manifestation of the inherent states of that system. State transition occurs in response to the inputs and other building blocks (Haimes, 2009).

Several attempts to merge system design and risk management have been made. Among these are: (1) CORAS (Fredriksen et al., 2003), a UML-based "practical framework, exploiting methods for risk analysis, semiformal methods for object-oriented modeling, and computerized tools, for a precise, unambiguous, and efficient risk assessment of *security critical systems*"; (2) Business process risk integrated modeling (Sienou et al., 2008), (3) Quantitative risk assessment for component-based systems (Grunske & Joyce, 2008); and (4) RiskM – a multi-perspective method for IT

risk assessment (Stricker et al., 2010), for the integration of IT risk consideration, assessment and mitigation, into business process modeling and execution.

Current modeling frameworks have several limitations, including: (1) Overemphasis on IT, and lack of support for general systems (real-time, C^4I systems, avionics systems, electro-mechanical systems, etc.), (2) Exaggerated occupation with physical and IT security; (3) Lack of model formalism, automated processing capabilities, consistency verification, and hierarchical analysis; and (4) Focus on logical cause-and-effect analysis, and insufficient support of quantitative analysis and assessment, probabilistic modeling, and uncertainty-related properties like hardware reliability.

Following this gap analysis, we propose ROSE – Risk-Oriented Systems Engineering – a new framework for risk-integrated systems modeling. ROSE's primary goals are: (1) Supporting general systems modeling and analysis, integrated with risk modeling, as well as various types of risk; (2) Supporting the system throughout its lifecycle, both during the developmental phase and the operational phase, with tools for risk-integrated system design, system configuration, control and management; and (3) Facilitating a flexible infrastructure which enables quick implementation and enhancement, and, eventually, (4) Increasing the ability to identify and mitigate risks.

Our framework builds on Object-Process Methodology – OPM (Dori, 2002), a structured conceptual modeling framework with a bimodal textual and visual representation. OPM is currently in the process of becoming an ISO standard, and an underlying framework for process and system modeling within ISO standards. OPM's simple yet robust notation enables the addition of modeling layers on top of, and in sync with the core system model, thus providing for coordinated and consistent multi-layered modeling. A risk model may be easily added as such a modeling layer.

The rest of this paper is organized as follows: Section 2 provides a brief description of our underlying methodology – OPM, and the reasons we chose this framework. In Section 3 we develop and present ROSE, our risk-oriented system configuration and design modeling theory and methodology. Section 4 presents a simple example of implementation of our framework during both design and operation of a shoulder missile defense system for commercial aircraft. Section 5 summarizes this paper, and delineates directions of future and complementary research.

2 Conceptual Modeling with OPM

Object Process Methodology, OPM (Dori, 2002) is a holistic, integrated approach to the design and development of systems in general and complex dynamic systems in particular. OPM integrates the structural and procedural views of a system into one view, and uses a minimal set of symbols. OPM aspires to modeling with detail evolution, rather than aspect evolution, as demonstrated by conceptual modeling languages like UML and SysML.

OPM building blocks are Objects and Processes. Objects are things that exist and can be stateful (i.e., have states). Processes transform objects: they generate and consume objects, or affect stateful objects by changing their state. These building blocks are

connected by Links of two types: structural and procedural. Structural links specify relations between objects, or between processes. Conversely, procedural links connect processes with objects or object states. OPM support the designation of entities as systemic or environmental, and as physical or informatical.

An OPM model consists of a set of hierarchically organized Object-Process Diagrams (OPDs). The hierarchical structure alleviates system complexity, through three mechanisms: (1) Unfolding and Folding structural hierarchies; (2) Zooming into or out of the inner details of things, and (3) Expressing or Suppressing the states of objects. Each OPD is obtained by in-zooming or unfolding of an object or process in its ancestor OPD. The graphical representation of an object is a rectangle, while a process is represented by an ellipse. Object states are represented by round-angle rectangles ("rountangles") within the owning object. The OPD hierarchical structure is accompanied by a corresponding set of structured textual model descriptions, written in Object-Process Language (OPL). OPL consists of automatically generated sentences in a subset of English edited in response to each user graphic editing. The textual formulation is equivalent to the graphical view, allowing for bimodal textual and visual description and understanding of the model.

OPM was selected as the basis for the ROSE framework for its following advantages: (1) Unification of the static-structural and dynamic-procedural aspects, using a single diagram type, at varying levels of detail, reduces clutter and incompatibilities even in highly complex systems. (2) Inherent complexity management, of divide and conquer by level of detail through its recursive seamless refinement-abstraction mechanisms. (3) A combination of semantically equivalent graphical and textual views makes OPM appealing to both sides of the human brain, so to speak. OPM models are quickly understood by professionals and practitioners. (4) Inherent capability to extend the core system model to additional aspects, while maintaining full coordination with the core model, as well as the capability to generate meta-models, i.e., generic, multi-purpose models and patterns, which can later be instantiated and adapted for specific systems and problems. (5) A freely available CASE tool – OPCAT, which implements almost all OPM concepts and allows fast adaptation and implementation. (6) OPM is currently in the process of becoming an ISO standard and a basis for ISO enterprise standards. This enables accelerated dissemination of OPM as a basis for enterprise modeling in general and risk modeling in particular.

3 Risk-Oriented Systems Engineering – ROSE

In this section, we present the ROSE framework. OPM serves as an enabling framework and language, as well as a meta-modeling framework for the theoretical and methodological aspects. The topmost diagram in the OPD hierarchy captures the entire essence of the system, provides an initial clear understanding of the problem or system discussed, and serves as an anchor for additional modeling. In addition, the OPM model captures both the system of interest in its environmental context. In our case, the primary process is simply the system's "Lifecycle". An OPD for SD-0 of our meta-model is illustrated in Figure 1. The textual OPL description follows in Figure 2.

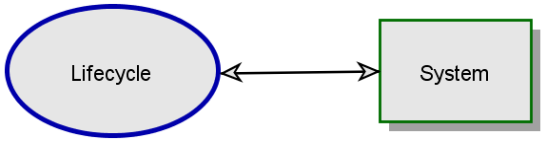

Figure 1. SD-0 - a general context of sysem lifecycle – OPD

```
0.   SD-0
0.1.   System is physical.
0.2.   Lifecycle affects System.
```

Figure 2. SD-0 – OPL for the topmost system diagram in Figure 1

We are now in-zooming into "Lifecycle", and extending SD-0 in SD-1. "Lifecycle" comprises two primary sub-processes: the developmental phase, "Project", and the operational – "Operation". Risk Management, is defined as a sub-process of both "Lifecycle" phases. SD-1's OPD is illustrated in Figure 3. The OPL text follows in Figure 4. Note, in both OPD and OPL, that "Project" is not linked to "System" in the same manner "Operation" is linked to "System: "*Project yields System*" (1.6), while "*Operation requires System*" (1.11).

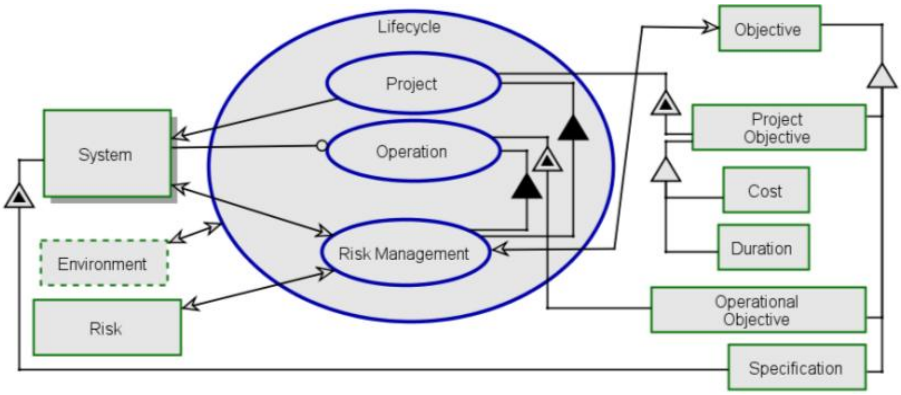

Figure 3. SD-1 - a closer look into sysem lifecycle and risk management - OPD

```
1.   SD-1 Lifecycle
1.1.   System exhibits Specification.
1.2.   Specification is an Objective.
1.3.   Lifecycle affects Environment.
1.4.   Lifecycle consists of Project, and Operation.
1.5.   Project and Operation consist of Risk Management.
1.6.   Project yields System.
1.7.   Project exhibits Project Objective.
1.8.   Project Objective and Operational Objective is an Objective.
1.9.   Cost is a Project Objective.
1.10.  Duration is a Project Objective.
1.11.  Operation requires System.
1.12.  Operation exhibits Operational Objective.
1.13.  Risk Management affects Risk, Objective, and System.
```

Figure 4. SD-1 - OPL

The OPDs at the next level down zoom into each process in the primary process. The SDs are designated hierarchically as SD-1.1 for "Project" (Figure 5a), SD-1.2 for "Operation" (Figure 5b), and SD-1.3 for "Risk Management" (Figure 5c). Excerpts from the corresponding OPL texts, referring to system design and configuration management, are brought in Figure 6.

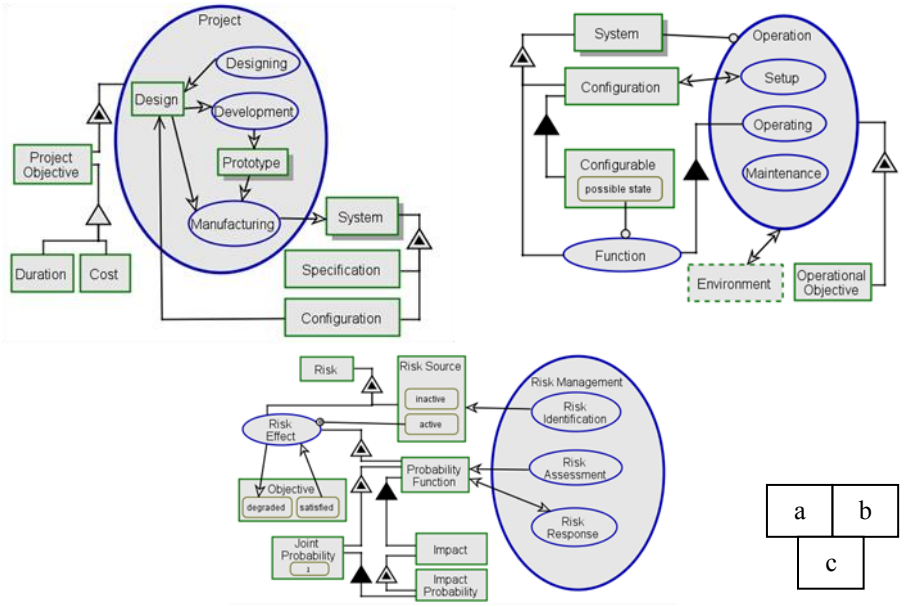

Figure 5. SD-1.1 – Project (a), SD-1.2 Operation (b), Risk Management (c)

```
2.   SD-1.1 Project
2.1.   System exhibits Configuration.
2.2.   Configuration relates to Design.
3.   SD-1.2 Operation
3.1.   Operation consists of Setup, Operating, and Maintenance.
3.2.   Setup affects Configuration.
3.3.   Configuration consists of many Configurables.
3.4.   Configurable can be possible state or alternative state.
3.5.   System exhibits Function.
3.6.   Function requires possible state Configurable.
3.7.   Operating consists of Function.
4.   SD-1.3 Risk Management
4.1.   Risk Management consists of Risk Identification, Risk
       Assessment, and Risk Response.
4.2.   Objective can be satisfied or degraded.
4.3.   Risk exhibits Risk Source, as well as Risk Effect.
4.4.   Risk Source can be active or inactive.
4.5.   Risk Source triggers Risk Effect when active.
4.6.   Risk Effect exhibits Probability Function.
4.7.   Probability Function exhibits Joint Probability.
4.8.   Joint Probability is 1.
4.9.   Joint Probability consists of many Impact Probabilities.
4.10.  Probability Function consists of many Impacts.
```

```
4.11.  Impact exhibits Impact Probability.
4.12.  Risk Effect changes Objective from satisfied to degraded.
4.13.  Risk Identification yields Risk Source.
4.14.  Risk Assessment yields Probability Function.
4.15.  Risk Response affects Probability Function.
```

Figure 6. SD-1.1, SD-1.2, and SD-1.3 – OPL excerpts

We are now in a position to formulate the meta-model which reflects the mutual effects of risk management on design and configuration. Indeed, risk management involves intricate subtle tradeoffs among contradicting objectives and fine balancing actions to maintain stability and overall Pareto-optimal satisfaction of the system within its environment. The meta-model is illustrated in Figure 7. The corresponding OPL supplement is described in Figure 8. This textual description is the essence of our methodology, as it guides the both the systems analysts and the risk analysts in integrating risk into system model and into configuration considerations.

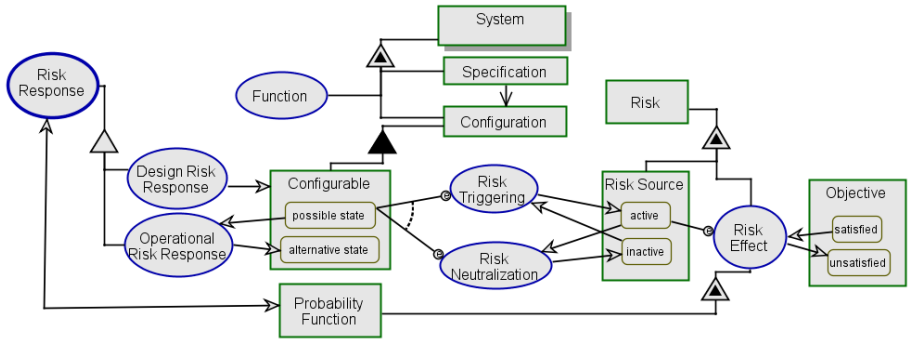

Figure 7. Risk Response through design and configuration decisions - OPD

```
5.   SD-1.2.3 Risk Response
5.1.   Design Risk Response is Risk Response.
5.2.   Design Risk Response yields Configurable.
5.3.   Operational Risk Response is Risk Response.
5.4.   Operational Risk Response changes Configurable from possible
       state to alternative state.
5.5.   Configurable triggers either Risk Triggering or Risk
       Neutralization when at possible state.
5.6.   Risk Effect exhibits Probability Function.
5.7.   Risk Neutralization changes Risk Source from active to inactive.
5.8.   Risk Triggering changes Risk Source from inactive to active.
```

Figure 8. SD1.2.3 – Risk Response – OPL

4 Example: Shoulder-Missile Defense System for Airliners

In this section we use a case study of a shoulder missile defense system designed for commercial airliners to illustrate ROSE in the design and configuration of the system. During the last decade, the threat of shooting down a commercial airliner with a simple and easily obtainable shoulder missile, by various terrorist organizations, has

dramatically increased. The Aerospace & Defense industry has tackled this challenge by equipping airliners with means to avoid or neutralize such a threat, bearing in mind that the pilot is not necessarily well-trained for such an action, and that the safety and welfare of the passengers and crew on board are of utmost importance.

The system described here is an imaginary, conceptual-level system, which has the ability to react on its own to identified threats, using one of several available countermeasures: a maneuver of the aircraft to avoid hit by the shoulder missile, attacking the shoulder missile with some weapon, like a missile or laser cannon, or disrupting the shoulder missile's capability by decoy scattering or radio signal transmission. The system supports pilot intervention and manual control, according to the level of hostility in the area, the availability of the different countermeasures on the aircraft, i.e., its countermeasure configuration, and the discretion of the pilot in case of an emergency, including the ability to avoid system activation in case of false alarm or threat infeasibility.

The basic system model is illustrated in SD-0 (Figure 9) and SD-1 (Figure 10). The primary process, Aircraft Defense, consists of three sub-processes: Threat Identification, Reaction, and Reporting.

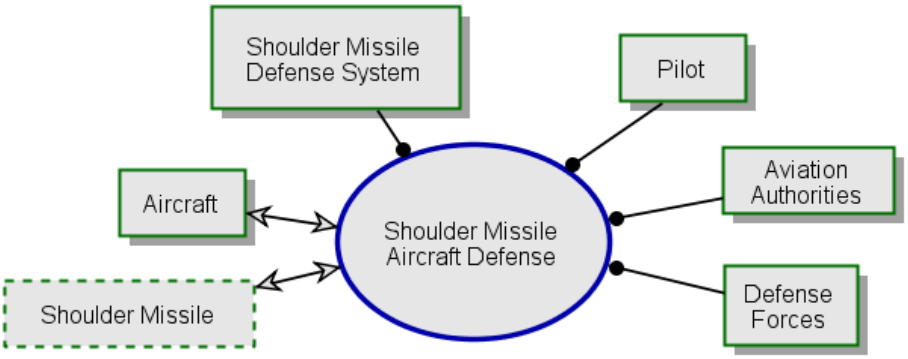

Figure 9. SD-0 – Shoulder Missile Defense System

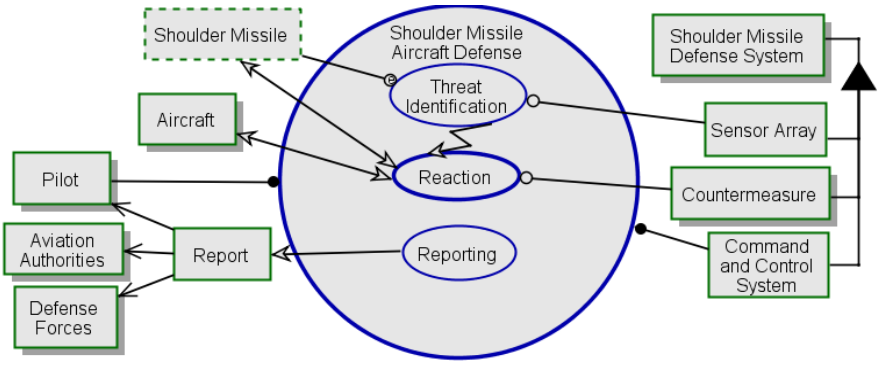

Figure 10. SD-1 – Shoulder Missile Defense System

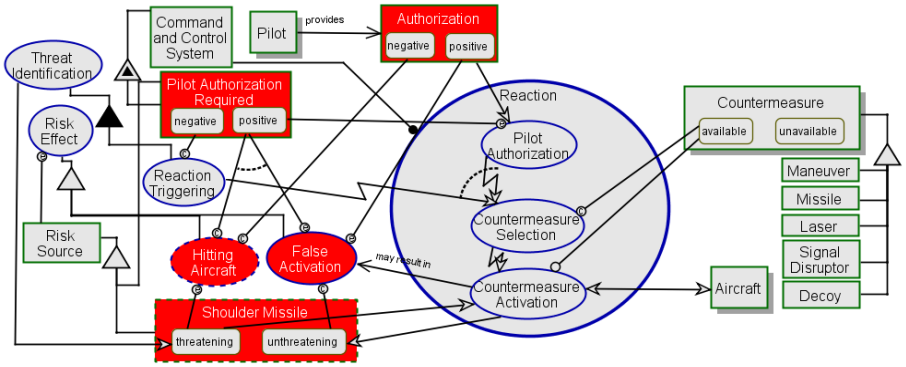

Figure 11. SD-1.2 – Shoulder Missile Defense System – Reaction Process

```
3.   SD-1.2 Reaction
3.1.   Threat Identification consists of Reaction Triggering.
3.2.   Shoulder Missile can be threatening or unthreatening.
3.3.   Threat Identification yields threatening Shoulder Missile.
3.4.   Reaction consists of Pilot Authorization, Countermeasure
       Selection, and Countermeasure Activation.
3.5.   Pilot Authorization Required can be positive or negative.
3.6.   Pilot Authorization Required triggers Pilot Authorization when
       positive.
3.7.   Pilot provides Authorization.
3.8.   Authorization can be positive or negative.
3.9.   Pilot Authorization invokes Countermeasure Selection.
3.10.  Reaction Triggering occurs if Pilot Authorization Required is
       negative.
3.11.  Reaction Triggering invokes Countermeasure Selection.
3.12.  Countermeasure Selection occurs if Countermeasure is available.
3.13.  Countermeasure Selection invokes Countermeasure Activation.
3.14.  Countermeasure Activation requires available Countermeasure.
3.15.  Countermeasure Activation affects Aircraft.
3.16.  Countermeasure Activation changes Shoulder Missile from
       threatening to unthreatening.
3.17.  Hitting Aircraft is Risk Effect.
3.18.  Shoulder Missile is a Risk Source.
3.19.  Pilot Authorization Required is a Risk Source.
3.20.  Authorization is a Risk Source.
3.21.  Hitting Aircraft may occur if Authorization is negative and
       Pilot Authorization Required is positive.
3.22.  Hitting Aircraft requires threatening Shoulder Missile.
3.23.  Shoulder Missile may trigger Hitting Aircraft when threatening.
3.24.  False Activation is Risk Effect.
3.25.  Countermeasure Activation may result in False Activation.
3.26.  Pilot Authorization Required may trigger False Activation when
       positive.
3.27.  Authorization may trigger False Activation when positive.
3.28.  False Activation occurs if Shoulder Missile is unthreatening.
3.29.  False Activation requires positive Pilot Authorization Required
       and positive Authorization.
```

Figure 12. SD-1.2.3 – Risk Response – OPL

In this example, we focus on the Reaction process (OPD in Figure 11, OPL in Figure 12), and on two risks: a shoulder missile may hit the aircraft, and the system may trigger false activation due to false positive identification of unthreatening objects. The system's support of manual operation is represented by a Configurable indicating if pilot authorization is required. The system may fail to react appropriately if the pilot does not authorize reaction against a real threat, or does authorize reaction against a falsely identified threat.

5 Summary

Systems are becoming highly integrated and interconnected, boundaries among business areas become fuzzy and fade out, system lifecycle is shortened and version cycles become more frequent. As these processes accelerate, it is vital to support the system's lifecycle with a suitable risk management methodology. Various risk scenarios arise during the design phase and coped with during the operational phase. Yet, the lack of documentation and of appropriate alignment and coordination with design and configuration decisions makes it difficult to conduct thorough risk identification, analysis, mitigation, and monitoring. The main reason for this is the fact that these two processes, namely development/management and risk management, are disparate and not integrated.

To meet this challenge, we have presented ROSE – Risk-Oriented Systems Engineering – a new approach to the integration of risk modeling and its consideration into the process of system design, configuration setting, and operational setup and activation. System engineers and risk analysts alike have been accustomed to practicing well-rooted methodologies, perceptions and working traditions. This poses a great challenge of bridging the gap between these two types of practitioners, which is mandatory in order for today's increasingly complex and multidisciplinary systems to thrive in a risk-plagued world.

This paper is a part of a broader research on conceptual modeling of systems in general and enveloping aspects like Risk Management in particular. Two goals guided this work. The first was to promote understanding and awareness of the importance of integrating system modeling and risk modeling by both systems analysts and risk analysts. Our second goal was to provide the means to model and understand Risk Management in general and more specifically in the context of design and configuration decision making. In the larger scope of this research, we approach Risk Management not just as a concept, but as an actual embodiment that plays an important role during the transition from the system's project lifecycle phase to the operation lifecycle phase. Throughout its lifecycle, the risk model is integrated into and synchronized with the system model, so as the system model evolves, so does the risk model. Along this line of thought, we are currently researching Lifecycle Risk Modeling and Management, including theoretical aspects, conceptual modeling with OPM, Risk Management extensions to OPM, and implementation of our methodology for various real-life applications.

References

Bedford T., Cooke R. (2001), *Probabilistic Risk Analysis: Foundations and Methods*, Cambridge University Press

Chapman C., Ward S. (2003), *Project Risk Management: Process, Techniques and Insights,* University of Southampton, John Wiley and Sons, 2nd Edition

Cooke R. (1991), *Experts in Uncertainty: Opinion and Subjective Probability in Science*, Oxford University Press

Diaz C.A. (1998), Product Re-Configurability and Product Introduction, *Concurrent Engineering September, vol. 6 no. 3 pp. 172-177*

Dori, D. (2002), *Object-Process Methodology – A Holistic Systems Paradigm,* Springer Verlag, Berlin, Heidelberg, New York, 2002.

Grunske, L., Joyce, D. (2008), Quantitative risk-based security prediction for component-based systems with explicitly modeled attack profiles, *Journal of Systems and Software, Volume 81, Issue 8, Pages: 1327-1345, ISSN: 01641212, DOI: 10.1016/j.jss.2007.11.716*

Haimes, Y.Y. (2008), Models for Risk Management of Systems of Systems, International Journal of Systems of Systems, Vol. 1, No. 1/2, pp. 222-236

Haimes, Y.Y. (2009), *Risk Modeling, Assessment and Management*, John Wiley and Sons, 3rd edition

Hoffman D.G. (2002), *Managing Operational Risk: 20 Firmwide Best Practice Strategies, Wiley Science*, John Wiley & Sons, Inc. New York

INCOSE (2004), *INCOSE Systems Engineering Handbook*, INCOSE-TP-2003-016-02, Version 2a, 2004, Version 3.0, 2009.

ISO/IEC/IEEE (2006), Systems and Software Engineering – Life Cycle Processes – Risk Management, ISO/IEC/IEEE Technical Publication 16085:2006, 2nd Edition

Krishnan, V., Ulrich, K.T. (2001), Product Development Decisions: A Review of the Literature, *Management Science, Vol. 47, No. 1, pp. 1-21*

NASA (1995), *NASA Systems Engineering Handbook*, NASA-SP-6105. SP-6105, 1995.

F. Redmill, M. Chudleigh, and J. Catmur (1999), *System Safety: HAZOP and Software HAZOP*, Wiley, Chichester, England.

Sharon, A. (2010), *A Unified Product and Project Lifecycle Model for Systems Engineering, Ph.D. thesis dissertation*, Technion – Israel Institute of Technology, March 2010

Sienou A., Lamine E., Pingaud H. (2008), A Method for Integrated Management of Process-risk, *Proceedings of the First International Workshop on Governance, Risk and Compliance* - Applications in Information Systems - GRCIS 2008, p. 16-30

Stoneburner G., Goguen A., Feringa A. (2004), Risk Management Guide for Information Technology Systems, NIST, Special publication 800-30 Rev. A

Strecker, S., Heise, D., Frank, U. (2010), RiskM: A multi-perspective modeling method for IT risk assessment, *Information Systems Frontiers, Volume 13, Number 4, pp. 595-611*