Streamlining Product Lifecycle Risk Management Through an OPM-Based Risk Modeling Framework

Y. Mordecai and D. Dori

Faculty of Industrial Engineering and Management

Technion – Israel Institute of Technology, Haifa 32000

ABSTRACT

Risk Management is common to projects and operations, the two major phases in the product's lifecycle, but is practiced independently and almost disconnectedly in these two domains. Unfortunately, current risk management methodologies hardly address the problem of risk transition from projects to operations, during the product's transition from the former phase to the latter, or vice versa. Project risk management is mostly concerned with risk reduction between project phases, e.g., design and development, while operational risk management usually assumes a given operational state, and ignores project risk effects. The problem worsens when project stakeholders have little motivation, and often some reluctance, to provide their operational counterparts with project risk information. In these cases, product beneficiaries are forced to spend significant time and resources, and to experience critical faults, in order to rediscover root problems and adverse effects that product engineers are usually aware of. Stakeholders on both sides have begun to acknowledge the transition of risks from the project phase in the product's lifecycle to the operational phase. Consequnetly, they are starting to collaborate in the streamlining and mitigation of these risks. We present a new overarching Lifecycle Risk Management framework, which consists of a structured risk modeling method, based on Object-Process Methodology (OPM), a structured conceptual modeling framework with a bimodal textual and visual representation. The new framework provides the means to model, analyze, mitigate, and manage risk migration between lifecycle phases.

Keywords: Lifecycle Risk Management (LRM), Object-Process Methodology (OPM), Risk Modeling, Conceptual Modeling, Project Risk Management, Operational Risk Management, Risk Migration, Risk Propagation, Risk Transformation.

1. INTRODUCTION

Product lifecycle consists of two major phases: the project phase and the operational phase. In the forrmer, the product is conceived, designed, developed, tested, produced and provided to the client. In the latter, the product is being used, operated, maintained and eventually disposed of. Both phases exhibit risk – the possible adverse effect on assets and objectives due to uncertain events. Risk Management is a common and essential practice in these two lifecycle phases, but risk analysts working in these two domains have little in common and share little information, if any.

Project Risk Management (PRM) aims at reducing the probability of occurrence of risks as well as the impact of risks, should they occur, at the project-, program-, portfolio- and organization-level (PMI, 2006; Chapman and Ward, 2003). PRM assists in channeling project progress in desired directions of its three primary objectives: to complete the project on schedule, to avoid cost (or budget) overruns, and to satisfy technical specification (spec) and quality requirements. PRM is an important activity in Project Management, but it usually ignores the post-delivery phase in the product's lifecycle.

Operational Risk Management (ORM) is concerned with assuring such objectives as reliability, safety, stability, security, functionality, availability, and business continuity. (the so-called "ilities", due to the common suffix of many of them) of organizations, facilities, processes, machines, products, and services engaged in operational settings, and exposed to continuous risk (Hoffman, 2002; Haimes, 2009). Organizations are encouraged to construct and execute contingency plans and procedures to handle such risks. Haimes (2009) distinguished ordinary, routine risk events, from rare events. Routine risk events, such as machine failures, traffic jams, bad weather, or competition, are mostly common, forseeable, and often absorbable. Unique risk events, such as earthquakes, floods, industrial espionage, and information theft, are critical and even disastrous, unforseeable and often unidentifiable. They are extremely more difficult to assess, prepare for, and respond to.

The lack of similarity of project objectives and operational objectives leads to estrangement between them, although these two objective types usually pertain to the same product along its lifecycle, or to the same organization in different business contexts. In spite of the seemingly minimal overlap, unmitigated project risks turn to operational risks, or generate them. In many cases, the continuous evolution of a product, while installed and operational in the client's premises, results in the constant evolution and emergence of development risks affecting the operational environment, and operational risks affecting the development one. Project risk managers focus on risk reduction within the project, e.g., from design to development or between development cycles. Operational risk managers usually cope with risks in systems in a given operational state, and disregard project risk effects. In many cases, project stakeholders have little in common with their operational counterparts, and hence have little motivation, and often some reluctance, to provide the latter with project risk information. Thus, product beneficiaries are forced to spend significant time and resources, and to experience critical faults or even disastrous consequences, in order to rediscover root problems and adverse effects that product engineers are usually aware of.

Unfortunately, neither PRM nor ORM provides the means to handle the transition or mutual effect of risks between phases. Moreover, the integration of PRM and ORM is not sufficiently addressed in the literature, except for special cases, in which product development and product operation are deeply intertwined, such as space missions (Cornford et al., 2003; Tralli, 2003; Haimes, 2009), nuclear reactors (Cooke, 1991) and oil platforms (Yang et al., 2009). Integrated risk management models for such unique and complex environments, which practice projects and operations simultaneously, mostly rely on similarities stemming from the shared business context, rather than the generic understanding of risk evolution and transition from the project phase to the operational phase. The proximity, and even uniformity, of project stakeholders and operational stakeholders in such settings further simplifies matters and reduces the opposition to risk information sharing. In practice, agile organizations, working in a system-centric approach, facilitate integrated multi-mission teams supporting a product along its entire lifecycle. In this case, the same person may support an existing client in the morning and work on modifications of a special version for a new client in the afternoon. The knowledge and experience acquired during operational support affects product development decisions, and vice versa.

The recent paradigm shift which adopts a holistic, lifecycle aware approach in various disciplines, has recently increased the awareness to the effects we describe. Stakeholders on both sides are acknowledging the transition and mutual effect of risks from the project phase in the product's lifecycle, to the operational phase, and seeking methods for collaboration in the streamlining and mitigation of these risks. With the increasing demand and emergence of challenges in this field, it is necessary to develop a body of knowledge on Lifecycle Risk Management (LRM), which encompasses and integrates project risk mangement (PRM) and operational risk management (ORM). Using LRM, we will be in a position to better address problems of risk transition and mutual effect among lifecycle phases. This would enhance and improve currently available risk identification, analysis and mitigation capabilities.

With this in mind, our three primary goals in this paper are the following:

A. Defining lifecycle risk management (LRM).

B. Laying down the foundations of LRM modeling and analysis.

C. Developing effective LRM mechanisms to handle lifecycle risk issues and problems.

Our framework consists of Object-Process Methodology (OPM), 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 actual system model, thus providing for consistency and improved collaboration between the system model and the wrapping model, in this case a risk management model. OPM also allows the unified modeling of both product design and project management (Sharon, 2010), which is essential for the integration of risk management models supprting these two phases.

The rest of this paper is organized as follows: Section 2 provides a description of Lifecycle Risk Management (LRM). Section 3 provides a brief description of our underlying methodoloy – OPM, and the reasons we chose this framework. In Section 4 we develop and present our Lifecycle Risk Management theory and methodology. Section 5 discusses the modeling of specific LRM problems, e.g. risk migration, risk propagation and risk transformation. Section 6 summarizes this paper, and depicts future research and extensions.

2. LIFECYCLE RISK MANAGEMENT (LRM)

We define Lifecycle Risk Management (LRM) as a holistic risk management approach, which focuses on the trade-off of risk between the two main lifecycle phases – the project phase and the operational phase, and bridges the gap between project risk management (PRM) and operational risk management (ORM). We realize that it is essential to consider the risk aspects in the functional characteristics of the product, as well as in the activities generating the product and its characteristics, all within the same unified model. Sharon (2010) introduces Project-Product Lifecycle Management approach – PPLM, as a holistic framework for product development and management. Our work extends PPLM to risk management, utilizing product lifecycle-oriented analysis foundations, and extending them to risk management notation, modeling and understanding. Thus, LRM becomes an important aspect of Project-Product Lifecycle Management.

LRM's primary objective is to deal with risk aspects in the lifecycle level, not necessarily to consolidate or integrate the risk management processes in the two major lifecycle phases. Therefore, we focus on the problems and issues of the simultaneous development and operation of a product, the transition from one phase to another (which is possible in both directions), and the effects of the operational environment on product development, or vice versa. We define three types of such lifecycle effects: risk migration, risk propagation, and risk transformation. Each effect is described below.

Risk migration is the direct movement of the same risk effect from one phase to another. During risk migration, the quantitative attributes of the risk may change, but the nature of the risk persists. Risk migration occurs when the same risk source, or a couple of similar risk sources, applies to both the developmental and operational environments, or when the same or similar objectives are shared by the project environement and operational environment. For example, interfaces to legacy systems may hinder the successful development of a new system, and later, the successful operation of the new system in the presence of the legacy systems, even if the interfaces were developed according to spec. Another example pertains to cost saved during the product development phase, which translates to a direct operational cost, e.g., when a power-intense alternative was preferred to an economic one for a particular system component, the operational expenses are expected to be higher. Such decisions are made all the time, since the life-cycle-cost (LCC) measure is not always the exclusive determining factor.

Risk propagation is a continuous effect of project progress on operations. For example, if the project schedule is delayed and the product is not ready, the operational environment is working inefficiently or unsafely, or cannot begin providing a new service for which the product is critical. Such propagation is also possible in the opposite direction: constant overload of the operational environment due to operational risks does not allow new product manufacturing or installation. For example, extensive aircraft maintenance within the operational phase delays or postpones aircraft modification in the next genertion, which is part of the developmental phase.

Risk transformation is the conversion of a project risk effect to another operational risk effect, or vice versa, due to the way the product design applies to the operational environment. Consider, for instance, a medium-sized company providing monitoring and control software, including support to provided solutions. The software application provides detailed diagnostics on the client's computation resources, and thus makes the client increasingly dependent on this product for premature problem identification and avoidance. If the application is inactive, critical failures may evade the client's controllers and expose the client to undesired effects, such as security breaches, resource overload, denial of service, and lack of control on system resource allocation. Thus, the information security level of the organization is affected by the reliability of the monitoring and control application. The reliability of the software consists of the various performance specifications defined and implemented in the product during the development phase.

3. OBJECT-PROCESS METHODOLOGY (OPM)

The challenge in creating a holistic framework is to unify data from several domains through a systematic model-based approach, language and methodology. These have to be intuitive, simple and formal. The characteristics of the modelling notation we looked for included the ability to represent data in hierarchically organized diagrams, expressiveness of the notation for defining common LRM conceptual metamodels, formalism, and clear semantics for execution-based model simulation. We also realized that risk analysis must be conducted on top of the actual product/project model, and not in a different and independent framework, like several common risk analysis frameworks, such as Cause-and-Effect diagrams, Fault-Tree Analysis (FTA), and Failure Mode Effect Critical Anaylsis (FMECA) (Haimes, 2009).

There are several conceptual and visual/graphical modeling frameworks available for risk modeling. CORAS (Fredriksen et al., 2003), is a European Union funded project for risk modeling, whose objectives are to develop a practical framework, exploiting methods for risk analysis, semiformal methods for object-oriented modelling, and computerised tools, for a precise, unambiguous, and efficient risk assessment of security critical systems. CORAS is basically an adaptation of UML to risk modeling. It mostly focuses on IT Security, and its applications concentrate in telemedicine and e-commerce. It is effective mainly in the operational phase of the prodcut's lifecycle.

Process-Risk Modeling Language (Sienou et al., 2008) is an integrated process modeling and risk modeling framework, which deals mostly with business processes and their risk aspects. It is also more effective for operational settings but it is more suitable for business processes than systems and products.

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 comprises entities and links. The three entity types are objects, processes (both referred to as "things"), and states. 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. Objects and processes are of equal importance, as they complement each other in the single-model specification of the system. Links, which are the OPM elements that connect entities, are of two types: structural and procedural. OPM objects relate statically to each other via structural relations, graphically expressed as structural links. Objects can also be structurally related to each other by unidirectional or bidirectional tagged relations, similar to association links in UML class diagrams. Structural relations specify relations between any two objects, or any two processes. Conversely, procedural links connect a process with an object or an object's state to specify the dynamics of the system. Procedural links include transforming links, enabling links, and control links.

An OPM model consists of a set of hierarchically organized Object-Process Diagrams (OPDs) that alleviate system complexity. Each OPD is obtained by in-zooming or unfolding of a thing (object or process) in its ancestor OPD. One or more new things (objects and/or processes) can be specified within a thing in an OPD that was refined from a higher-level OPD.

The graphical representor of an object in OPD is a rectangular, while a process is represented by an ellipse. Object states are represented by round-angle rectangulars (often called "rountangulars"). There are some more representors and representation modifiers we will refer to when neessary throughout this paper.

OPM was selected as the basis for LRM due to its following features:

1. OPM is a visual methodology that incorporates the static-structural and dynamic-procedural aspects of a system into a unifying model, which is presented in its entirety using a single diagram type. This is achieved by treating both objects and processes as equally important things (entities). By using a single model at varying levels of detail, clutter and incompatibilities are likely to be avoided even in highly complex systems.
2. OPM is designed to express triggering events, guarding conditions, timing constraints, timing exceptions, and flow-of-control constructs. These features are the basic elements required for complex dynamic behaviors in general, and temporal exceptions in particular.
3. OPM has a unique approach for complexity management and reduction, through recursive seamless complexity management mechanisms (scaling, or abstraction/refinement). These mechanisms are (1) unfolding/folding, which is used for refining/abstracting the structural hierarchy of a thing; (2) in-zooming/out-zooming, which exposes/hides the inner details of things within its frame; and (3)

expressing/suppressing, which exposes/hides the states of an object. These complexity management mechanisms enable OPM to represent complex systems gradually.

4. OPM consists of two semantically equivalent modalities of the same model: graphical and textual. A set of interrelated Object-Process-Diagrams (OPDs) constitute the graphical model, and a set of automatically-generated sentences in a subset of English constitute the Object-Process Language (OPL). In the graphical-visual model, each OPD consists of OPM elements depicted as graphic symbols, while the OPD syntax specifies the consistent and correct ways by which those elements can be managed. Since the corresponding textual model is generated in a subset of English, it is immediately understood by domain experts, who need not learn any special language nor decipher cryptic code.
5. OPM allows us to extend the basic system model, to various additional and complementary aspects, on top of, and in full synchronization with, the system model itself. This feature is useful for several applications, such as project management, requirements management, and risk management.
6. OPM features a free CASE tool – OPCAT, which implements almost all OPM concepts and allows fast adaptation and construction of dedicated meta models.
7. OPM is currently in the process of becoming an ISO standard and a basis for ISO enterprise standards. When completed, this will enable accelerated dissemination of OPM as a basis for enterprise standards in general and for LRM in particular.

4. MODELING LIFECYCLE RISK MANAGEMENT WITH OPM

4.1 General Lifecycle Modeling

In order to create our first OPM-based risk model, we follow two important OPM guidelines: the meta-modeling and the hierarchical modeling. The first guideline, meta-modeling, is the definition of a generic, multi-purpose model, which depicts the theoretical infrastructure of the model. This model can later be instantiated and adapted for specific systems and problems.

The second guideline, hierarchical modeling, is the way to work with OPM in general, and with its CASE tool – OPCAT, in the most efficient and effective way. OPM aspires to modeling with detail evolution, rather than aspect evolution, as demonstrated by conceptual modeling languages like UML and SysML. We construct a hierarchy of system/process diagrams, in which each diagram is subject to its parent diagram, and oversees its child diagrams. The addition of elements in any level can bubble up to the topmost level, or permeate the deepest one.

The first and topmost diagram in our meta-model is labeled SD-0 (SD stands for System Diagram). Our SD-0 includes only one object and one process. The object is "Product", and the process is "Product Lifecycle". Figure 2 presents our SD-0. The corresponding textual description in OPL is short and somewhat trivial. It is presented in Figure 1.

Product Lifecycle affects Product.

Figure 1. Product Lifecycle - SD-0 – OPL

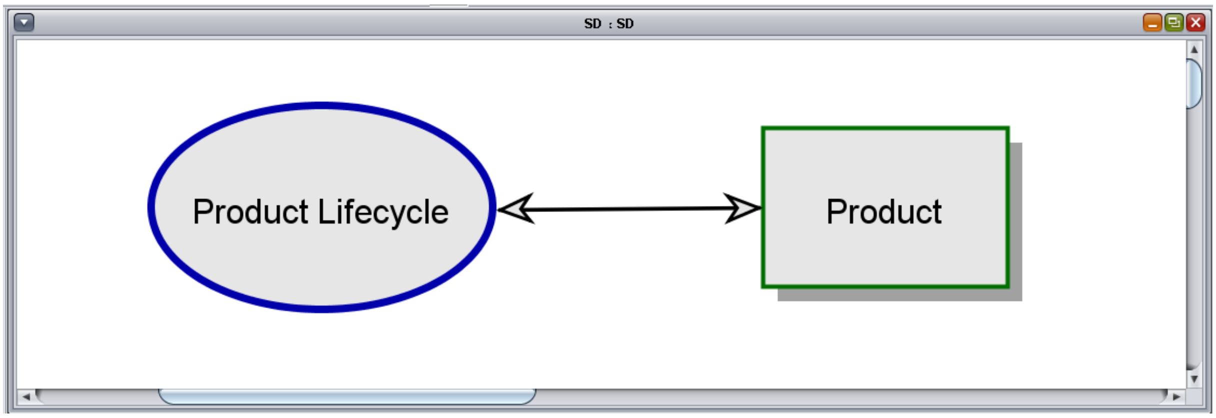

Figure 2. Product Lifecycle - SD-0 - OPD

The next step in OPM is the "In-Zooming" of the primary process, Product Lifecycle, in a diagram labeled SD-1, subject to SD-0, as displayed in Figure 3. We now have the capability to provide details about the internal structure and dynamics of the process, and denote the two primary lifecycle phases, the Project and the Operation, as sub-processes. We also define the Risk Management process as a sub-process of both of them. Note that while Project is initially considered to affect Product, we use a different link to express the concept that Product is instrumental for Operation. This distinction is important in this level in order to designate the roles of the two primary lifecycle phases, but may be refined in lower levels, in which it will also be possible for Operation to change or affect Product. The OPL description corresponding to SD-1 is provided in Figure 4.

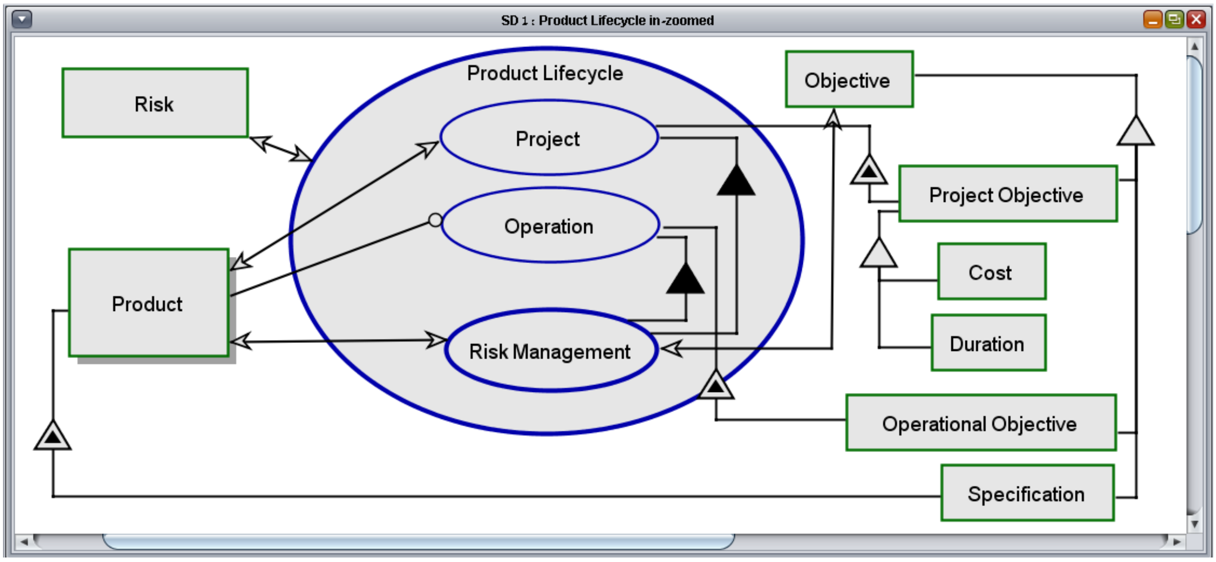

Figure 3. Product Lifecycle - SD-1 - OPD

Another important notion in SD-1 is the definition of the Objective hierarchy. Project, Operation, and Product, each have their own special objectives, as also modeled. Risk Management has an effect on Objective, since Risk has an effect on Objective. The Objectives of Project and Operation may be different but they are all modeled as Objectives, initially. We further refine and specify these objectives downstream.

Product is physical.
Product exhibits Specification.
  Specification is an Objective.
Cost is a Project Objective.
Duration is a Project Objective.
Product Lifecycle consists of Risk Management, Project, and Operation.
Product Lifecycle affects Risk.
Product Lifecycle zooms into Project, Operation, and Risk Management.
  Project exhibits Project Objective.
    Project Objective is an Objective.
  Project consists of Risk Management.
    Risk Management affects Objective and Product.
  Project affects Product.
  Operation exhibits Operational Objective.
    Operational Objective is an Objective.
  Operation consists of Risk Management.
  Operation requires Product.

Figure 4. Product Lifecycle - SD-1 - OPL

The next step includes in-zooming of each sub-process in SD-1. Since our focus in this paper is on Risk Management, this is a process we would like to in-zoom. By leaving Project and Operation in their generic form we are able to discuss such activities without loss of generality.

4.2 Risk Management Modeling

Risk Management takes shape in SD-1.1, shown in Figure 5. In this diagram we model both Risk and Risk Management. In the Risk model, we distinguish Risk Source from Risk Effect. Risk Source, *per se*, is not something risky, i.e. it does not necessarily incur risk to anything. When it might generate an adverse effect on objectives of special interest to us, it becomes a risk source. Stormy wheather, for instance, is not something of risk, but its effect may be, for instance, communication disruption, velocity reduction, wettening, overflowing, etc., which, in turn, may adversely affect our Objectives. Since this is a meta-model, we generically define Risk Source, but various objects, either systemic or environmental, may constitue Risk Sources, as demonstrated later. We also model several lifecycle risk phenomena, namely Risk Migration, Risk Propagation, and Risk Transformation, which we refer to in Section 5.

The auto-generated OPL text corresponding to SD1.1 is provided in Figure 6. Note the way OPL expresses the cardinality constraints used to make sure that the joint probability, which is the sum of all impact probabilities, equals 1.

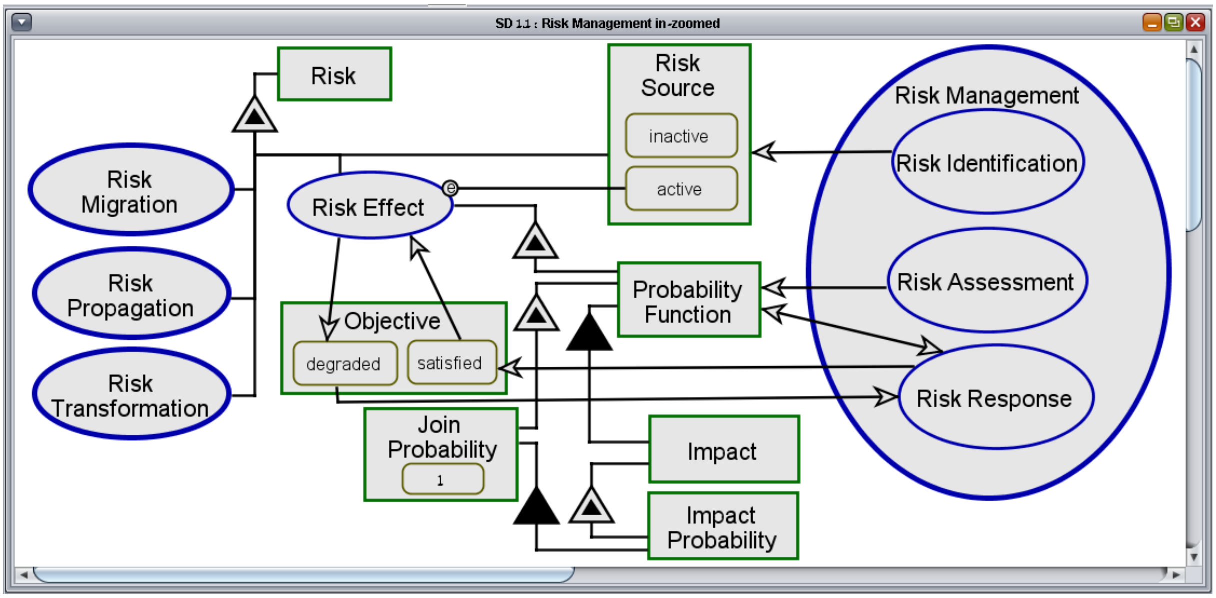

Figure 5. Risk Management – SD-1.1 – OPD

In the Risk Management process modeling, we focus on three primary Risk Management processes: Risk Identification, Risk Assessment, and Risk Response.

Risk Identification is the process in which Risk Sources are identified, or defined. In the model, as also expressed in OPL, Risk Indentification yields Risk Source, but, for our purposes, this means definition, rather than generation, since the Risk Source may exist, regardless of its identification.

Risk Assessment constructs the Risk Effect's Probability Function. Again, the process is responsible for subjective modeling and capturing of uncertainty, not for generating the *real* probability distribution of the event, which is unknown, and according to the Subjectivist school of thought (De-Finetti, 1974; Savage, 1972), non-existent to begin with.

Risk Response, which includes both planning and execution, affects both Probability Function and Objective, since it can change one Objective on the expense of another in order to respond to a risk (e.g., by waivering a performance requirement in order to meet schedule, or add budget in order to support additional features necessary for risk response). Note that our model enables continuous, discrete, or singular probability distribution, as Probability Function consists of one or more Impact values, each one of which exhibits a probability value. All probability values must, of course, sum to 1.

Objective can be satisfied or degraded.
Risk exhibits Risk Source, as well as Risk Effect, Risk Migration, Risk Propagation, and Risk Transformation.
Risk Source can be active or inactive.
Risk Source triggers Risk Effect when it enters active.
Risk Effect exhibits Probability Function.
Probability Function exhibits Join Probability.
Join Probability is 1.
Join Probability consists of many Impact Probabilities.
Probability Function consists of many Impacts.
Impact exhibits Impact Probability.
Risk Effect requires active Risk Source.
Risk Effect changes Objective from satisfied to degraded.
Risk Management consists of Risk Identification, Risk Assessment, and Risk Response.
Risk Management zooms into Risk Identification, Risk Assessment, and Risk Response.
Risk Identification yields Risk Source.
Risk Assessment yields Probability Function.
Risk Response affects Probability Function.
Risk Response changes Objective from degraded to satisfied.

Figure 6. Risk Management – SD-1.1 – OPL

5. MODELING LIFECYCLE RISK MANAGEMENT PHENOMENA

We now turn to the modeling of lifecycle risk phenomena mentioned in Section 1 above. The modeling draws on the meta-model presented in Section 4. This is also done in meta-modeling form, which can later be applied and instantiated to actual cases. We refer to these phenomena as processes exhibited by Risk, as shown in Figure 5, and textually described as: "Risk exhibits Risk Migration, Risk Propagation, and Risk Transformation".

5.1 Risk Migration

As explained, risk migration is, in general, the movement of the risk effect from the project domain to the operational domain, when the product shifts from the project phase to the operational phase, or resides simultaneously in both phases.

Risk migration is modeled as the same Risk Effect affecting both a Project Objective and an Operational Objective. The Risk Migration pattern is presented in Figure 7. It is fairly simple to model, as can also be understood from the simple OPL formulation: "Risk Effect affects Project Objective and Operational Objective".

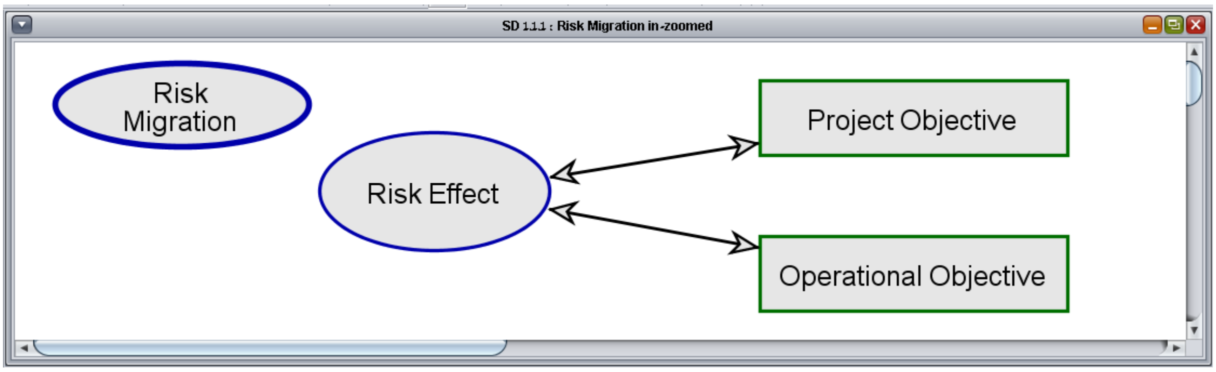

Figure 7. Risk Migration – OPD

5.2 Risk Propagation

Risk propagation is a continuous effect of project progress on operations. In this case, Project Objectives constitute Risk Sources, and have a special effect on Operational Objectives. This concept is modeled in Figure 8. The main principle which can also be seen in the OPL is that a Project Objective is a Risk Source, and that the Risk Effect triggered by the Risk Source affects an Operational Objective, or vice versa – an Operational Objective may constitute a Risk Source to a Project Objective.

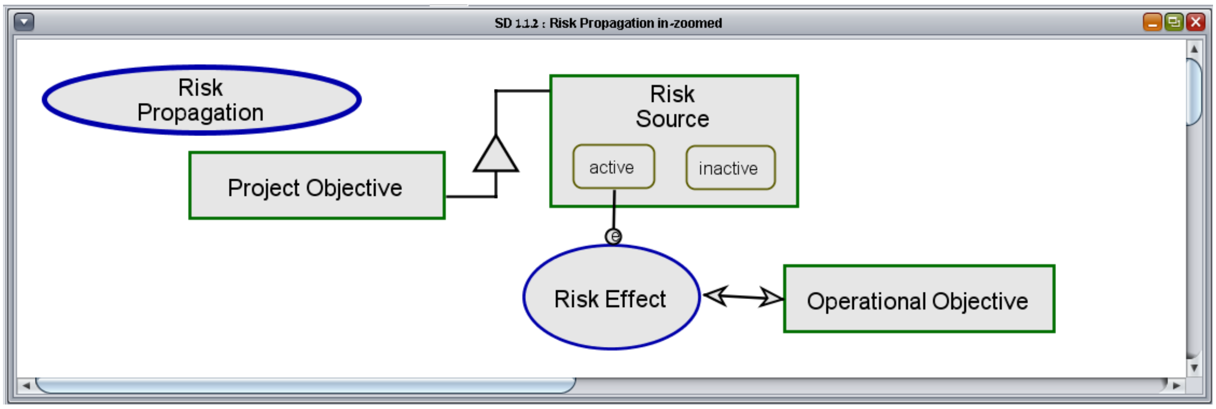

Figure 8. Risk Propagation – OPD

5.3 Risk Transformation

Risk transformation is the conversion of a project risk effect to an operational risk effect, or vice versa. In this process, a risk in the project domain is transformed to a risk in the operational domain. The operational risk is not the same as the project risk, but it is related to it. As opposed to Risk Migration or Risk Propagation, which are emergent and demonstared by the model, and not substanstial process, the transformation process should be defined as a part of the model, in order to understand its nature. Even if the transformation is not recognized in the first place, merely assuming and examining its possible existence between various risks in the actual model, is a useful risk identification technique. Modeling Risk Transformation is a bit tricky, as shown graphically in Figure 9, and textually in Figure 10.

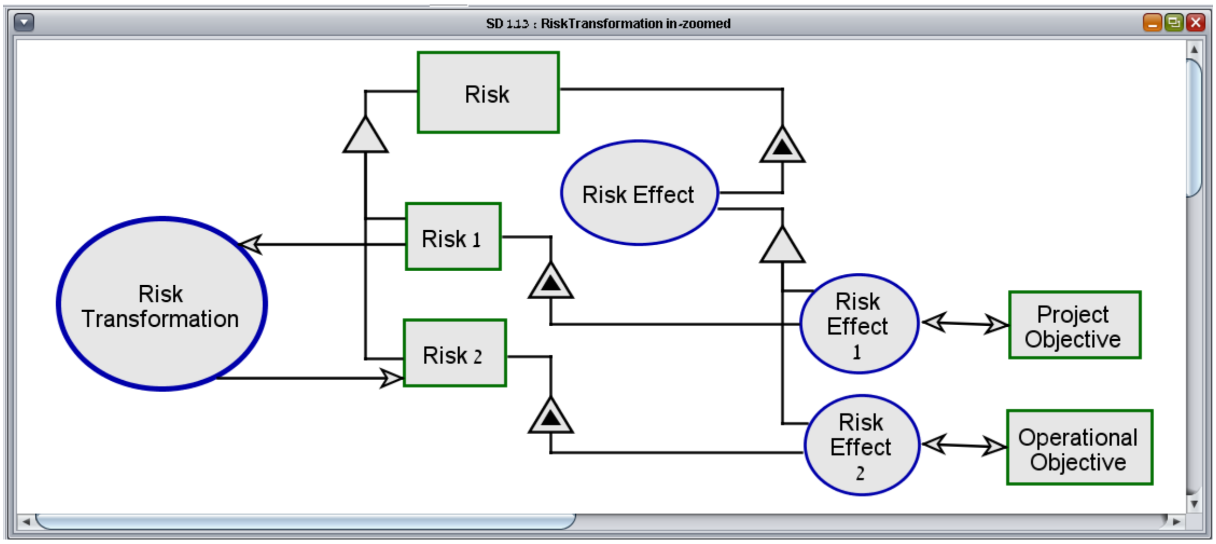

Figure 9. Risk Transformation – OPD

Risk exhibits Risk Effect.
Risk 1 is a Risk.
Risk 1 exhibits Risk Effect 1.
    Risk Effect 1 is Risk Effect.
    Risk Effect 1 affects Project Objective.
Risk 2 is a Risk.
Risk 2 exhibits Risk Effect 2.
    Risk Effect 2 is Risk Effect.
    Risk Effect 2 affects Operational Objective.
Risk Transformation consumes Risk 1.
Risk Transformation yields Risk 2.

Figure 10. Risk Transformation – OPL

6. SUMMARY

Although project risk management and operational risk management are commonly practiced independently, we believe that the integration of these two primary risk management processes poses one of the greatest challenges in current risk management research. As businesses become highly integrated and interconnected, as the boundaries among business areas decrease and fade out, as product lifecycle is shortened and product version cycles become more frequent, it is vital to support the entire business activity with a suitable risk management methodology. This is not easy, as each business discipline has well-rooted methodologies and perceptions for risk management, modeling and mitigation, and it is a great challenge to overarch the gaps among risk management practitioners, in order to provide relevant and necessary capabilities for today's world's complex multidisciplinary businesses.

In this paper, which is a part of a broader research on the conceptual modeling of systems in general, and enveloping aspects like Risk Management specifically, we tried to achieve two goals. Our first goal was to put Lifecycle Risk Management on the table, as a research problem, as a branch of Risk Management and as an overarching framework. Our second goal was to provide the means to model and understand Risk Management in general, and LRM in particular, not just as a concept or idea, but through its actual embodiments in the transition from projects to operations, or coexsitence thereof, and directly on top of the product's model, and not through a disconnected risk model, which is not synchronized with the constant evolution of the system in question.

We are currently conducting extensive research on LRM, including theoretical aspects, conceptual modeling with OPM, Risk Management extensions to OPM, and implementation of our methodology for various real-life applications.

7. REFERENCES

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*

Cornford S.L., Paulos T., Meshkat L., Feather M. (2003), Towards More Accurate Life Cycle Risk Management Through Integration of DDP and PRA, *NASA, Jet Propulsion Laboratory, California Institute of Technology*

de Finetti, B. (1974), Theory of Probability, *Wiley, New York*

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

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*

Fredriksen R., Kristiansen M., Gran B.A., Stølen Ketil, Opperud T.A., Dimitrakos T. (2002), The CORAS Framework for a Model-Based Risk Management Process, *Computer Safety, Reliability and Security - Lecture Notes in Computer Science, 2002, Volume 2434/2002, 39-53, DOI: 10.1007/3-540-45732-1_11*

PMI – Project Management Institute (2006), The Project Management Body of Knowledge – PMBOK, *PMI – Project Management Institute*

Savage G. (1972), The Foundations of Statistics, *Dover, New York. First published on 1954, John Wiley & Sons*

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*

Tralli, D.M. (2003), Programmatic Risk Balancing, *Proceedings of the IEEE Aerospace Conference, 2003, p. 2_775 - 2_784*

Yang G.Q., Zhang Chun-C.J., LIU Z.G. (2009), FPSO Lifecycle Risk Management, *Ship & Ocean Engineering, 2009, number 1*