

EMI: Engineering and Management Integrator

Michael Masin, Yael Dubinsky, Michal Iluz, Evgeny Shindin
and Abraham Shtub

Abstract The impact of systems engineering on program cost has been recognized for over a decade. From the very early stages, careful management of the relationships between the product design and the project plan is crucial to the success of any project that aims to deliver a defined product. Failure to closely manage the intricate web of resource constraints emanating from the two domains, the project scope and the product scope may lead to inadequate product performance or overruns in project schedule and budget. Identifying and managing the relationship between these two domains are at the heart of our challenge to combine project management (PM) and systems engineering (SE). We present a new approach, called EMI, which integrates SE and PM methodologies. These include the EMI mathematical foundation, implementation in architectural optimization and project management tools, and a detailed use case for development of the Doors Management System for commercial aircraft.

M. Masin (✉) · Y. Dubinsky · E. Shindin
IBM Research—Haifa, Mount Carmel, Haifa 31905, Israel
e-mail: michaelm@il.ibm.com

Y. Dubinsky
e-mail: dubinsky@il.ibm.com

E. Shindin
e-mail: evgensh@il.ibm.com

M. Iluz · A. Shtub
Technion—Israel Institute of Technology, Technion City, Haifa 34000, Israel
e-mail: iluzmichal@gmail.com

A. Shtub
e-mail: shtub@ie.technion.ac.il

1 Introduction

“All programs...shall apply a robust Systems Engineering approach that balances total system performance and total ownership costs...” [1]. Meeting the required system specifications is one of the main challenges for systems engineers, whether chief system engineers or system engineers of a specific discipline. One of their objectives is to meet budget and schedule goals for the development and manufacture of the system and its subsystems. The ultimate goal is to provide a high quality product on time and within budget. Systems engineering (SE) and project management (PM) are two tightly intertwined domains, as stated in the Handbook of Systems Engineering and Management [2]. In contemporary practice, system engineers relate to the development and manufacturing costs that can be measured in work hours. However, when considering technological approaches, system engineers may choose state-of-the-art solutions without fully considering schedule and budget implications and constraints. In other words, they sometimes focus more on the product scope and neglect the project scope. System engineers make product-domain decisions that directly influence the project-domain, in which the project manager is responsible for multi-criteria decision-making (MCDM) on a daily basis. Analytic and simulation decision support tools can be of great value in such an environment.

Starting with the pivotal work of Forsberg and Mooz [3], many papers describe the need or conceptual framework for integrating SE and PM (see, e.g., [4–8]). Our work shows how this integration can be accomplished. This paper presents EMI, short for Engineering and Management Integrator, which combines SE and PM methodologies. Our contribution is in developing a practical mathematical model that combines SE architectural optimization and PM planning tasks. As part of our work, we implement this method using decision support tools, define a holistic engineering and management methodology, and demonstrate it for a typical product development.

The paper is organized as follows. We first describe the EMI mathematical foundation and methodology. We then demonstrate the methodology in a typical industrial use case involving the development of the Doors Management System (DMS) in a commercial aircraft using two state-of-the-art tools. The first tool is the Architectural Optimization Workbench (AOW), which deals with the architectural optimization (AO) aspects. The second is the Project Team Builder (PTB), which deals with the PM aspects. We show the limitations that arise when each tool is used on its own and how using EMI to integrate these tools can overcome the limitations and improve the results for finding an efficient DMS architecture. Finally, we summarize and provide directions for future research.

2 EMI Mathematical Foundation and Methodology

2.1 Project Time Management for Architectural Optimization

While PM has many aspects [9], in developing EMI we focus on the integration of time management and selection of the best system architecture. The mathematical model closest to the settings of AO is the Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP) [10, 11]. In MRCPSP, project activities have several operational modes, each with its own duration and required set of resources. The activities have precedence constraints, and resources have a final capacity. An MRCPSP solution defines the mode in which each activity is executed and schedules the activities according to precedence and resource constraints. Solution procedures for MRCPSP include both heuristic and exact methods. In off the shelf MILP solvers, such as Cplex [12], the exact methods based on MILP formulation are capable of solving industrial size problems [13]. It is interesting to note, that the classical time-indexed formulation or its slight modification are usually the preferable options for solving real size problems with a few hundreds of periods [11].

We begin our methodology by adjusting MRCPSP to our AO needs. First and foremost, we synchronize the mode selection with the selected architectural components. Development projects are usually performed with preemptive schedules and part-time job intensity. This is especially true in matrix organizations, where technical units with domain expertise provide services to all running projects. Our model also supports variable period lengths. The total number of periods and consequently, model size, could be significantly reduced using longer period lengths for later periods, where having a detailed plan makes less sense. The model we developed, called AO-MRCPSP, is described below.

Sets: A —activities, IP_i —immediate predecessors of activity $i \in A$, M_i —modes of activity $i \in A$, R —resources, $P = 1 \dots T$ —periods, G —subsystem/components types. **Parameters:** w —the minimum work intensity if an activity is performed in a part-time, e —maximal extension of activity duration caused by preemption, p_t —duration of period $t \in P$, $T_{max} = \sum_{t \in P} p_t$ —time horizon, d_j —duration of mode j , r_{jk} —requirement of mode j for resource k , a_j —resource independent cost of mode j , h_i —subsystem/component type of activity i , n_j —subsystem/component type id of mode j , v_k —capacity of resource k , b_k —cost of resource k per time horizon. **Decision variables:** x_{ij} —binary variable for mapping activity $i \in A$ to mode $j \in M_i$, y_{jt} —continuous variable for mapping mode j to period $t \in P$, \tilde{y}_{it} —binary indicator that activity $i \in A$ is performed at period $t \in P$, s_{it} —binary wave variable that activity $i \in A$ started at period $t \in P$ or earlier, f_{it} —binary wave variable that activity $i \in A$ finished before period $t \in P$ or earlier, C_i —completion time of activity $i \in A$, C_{max} —completion time of the whole project, u_{kt} —utilization of resource k at period t , u_k —average utilization of resource k , A —total cost of activities, B —total cost of resources, D —total cost of project, q_o —type id of subsystem/component type $o \in G$.

$$\begin{aligned} \mathbf{AO} - \mathbf{MRCPSP} \quad & \text{Minimize} \{C_{max}, D\} \\ & \text{Subject to} \end{aligned} \quad (1)$$

$$\sum_{j \in M_i} x_{ij} = 1 \quad \forall i \in A \quad (2)$$

$$y_{jt} \leq x_{ij} \quad \forall i \in A, j \in M_i, t \in P \quad (3)$$

$$w \cdot \tilde{y}_{it} \leq \sum_{j \in M_i} y_{jt} \quad \forall i \in A, t \in P \quad (4)$$

$$\tilde{y}_{it} \geq \sum_{j \in M_i} y_{jt} \quad \forall i \in A, t \in P \quad (5)$$

$$s_{it} \leq s_{i,t+1} \quad \forall i \in A, t \in P | t < T \quad (6)$$

$$f_{it} \leq f_{i,t+1} \quad \forall i \in A, t \in P | t < T \quad (7)$$

$$s_{it} \geq \tilde{y}_{it} \quad \forall i \in A, t \in P \quad (8)$$

$$f_{it} \leq 1 - \tilde{y}_{it} \quad \forall i \in A, t \in P \quad (9)$$

$$\tilde{y}_{it} \leq f_{i't} \quad \forall i \in A, i' \in IP_i, t \in P \quad (10)$$

$$\sum_{j \in M_i} \sum_{t \in P} y_{jt} p_t \geq \sum_{j \in M_i} x_{ij} d_j \quad \forall i \in A \quad (11)$$

$$\sum_{t \in P} (s_{it} - f_{it}) p_t \leq e \sum_{j \in M_i} x_{ij} d_j \quad \forall i \in A \quad (12)$$

$$C_i = T_{max} - \sum_{t \in P} f_{it} p_t \quad \forall i \in A \quad (13)$$

$$C_{max} \geq C_i \quad \forall i \in A \quad (14)$$

$$u_{kt} = \frac{1}{v_k} \sum_{i \in A} \sum_{j \in M_i} y_{jt} r_{jk} \quad \forall k \in R, t \in P \quad (15)$$

$$u_{kt} \leq 1 \quad \forall k \in R, t \in P \quad (16)$$

$$u_k = \frac{\sum_{t \in P} u_{kt} p_t}{\sum_{t \in P} p_t} \quad (17)$$

$$B = \sum_{k \in R} u_k v_k b_k \quad (18)$$

$$A = \sum_{i \in A} \sum_{j \in M_i} x_{ij} a_j \quad (19)$$

$$D = A + B \quad (20)$$

$$\sum_{j \in M_i} x_{ij} n_j = q_{h_i} \quad \forall i \in A \quad (21)$$

$$x_{ij}, \tilde{y}_{it}, s_{it}, f_{it} \in \{0, 1\} \quad 0 \leq y_{jt}, u_{kt}, u_k \leq 1 \quad C_i, C_{max}, A, B, D \geq 0 \quad (22)$$

The total project duration and cost are common objective functions in Eq. (1). Moreover, any piecewise linear function of decision variables could be added to the set, for example, the utilization range of critical resources to smooth their usage. Constraints (2)–(3) ensure exactly one mode for each activity. Constraints (4) define minimal part-time intensity w if activity i is performed during period t , for example, 50 % to allow working on two activities at most. Note, only one y_{jt} can be positive. Constraints (5) connect mode continuous performance to an activity binary indicator at period t . Constraints (6)–(9) ensure correct behavior of wave functions when activity i starts and finishes (because of objectives and other constraints, s_{it} and f_{it} try to become 1 as late and as early as possible, respectively). Constraints (10) ensure precedence relation, allowing activity i to start only after all its predecessors have finished. This constraint could be relaxed to $\tilde{y}_{it} \leq f_{i',t+1}$ to allow simultaneous execution of an activity with its predecessors for rough resource allocation (e.g., in later long periods). Constraints (11) ensure activity i is performed long enough to be completed. Effectively, these constraints imply that the project should be completed before T_{max} . Constraints (12) restrict the total time activity i has started but not finished yet to e times its nominal duration d_j . Constraints (13)–(14) calculate activity and project completion time, respectively. Activity start time could be calculated similarly using s_{it} instead of f_{it} . Additional constraints could be added to the earliest and latest start and completion times. Constraints (15)–(17) calculate resource utilization and constrain it according to the available capacity. We allow variable resource capacity over periods in our implementation. Constraints (18)–(20) calculate the cost of resources, B , the cost of activities, A , and the total project cost, D , respectively. Constraints (21) relate the PM model with AO, ensuring that activities connected to some subsystem/component choose the same subsystem/component type as AO. Constraints (21) define domain of decision variables. AO-MRCPSP can incorporate time and resource buffers. Dummy activities without required resources (i.e., $r_{ik} = 0$ for all k) before integrating activities and before activities that request critical resources represent time and resource buffers, respectively.

The AO-MRCPSP formulation above has several beneficial properties. Allowing preemption and part-time intensity help relax regular MRCPSP constraints and remove the need for binary indicators per mode per period. In this setting, event-based formulations [14], usually most applicable for large projects, are less attractive since preemptions require more events. In addition, the model is not very

sensitive to the number of modes—most variables and constraint sets are per activity, not per mode. Probably the most important property of AO-MRCPSP is that it requires a relatively small number of periods, which is the most sensitive and problematic parameter of time-indexed formulation. Currently, standard optimization packages, such as Cplex [12] can handle up to a few hundreds of periods. If each period represents a week, our model optimizes a multi-year plan. During AO, a rough estimation is required with a lot of uncertainty regarding later stages of the project. The period lengths could be adjusted accordingly, further reducing the model size. For example, in our use case we apply bi-weekly periods.

2.2 AO-MRCPSP in Architectural Optimization

To incorporate the aspect of architectural optimization into EMI, we relied on two preliminary works. The first is the *concise modeling* [15] extension of SysML [16] to specify architectural alternatives and system constraints. Concise modeling combines regular SysML diagrams that define architectural topology (e.g., components multiplicity and connections rules), with associated data tables called *catalogs* for block subclasses and *inventories* for part multiplicities. Using concise modeling, AO assumes that all parts are *a priori* optional, and divides all attributes into either *parameters* determined by data tables or SysML model values, or *variables* optimized during AO. For example, when outlined by *catalog* stereotype, the RDC block has an associated catalog table listing several RDC options and their relevant parameters (weight, power, etc.). These options could include different technologies, such as optical or copper cables, that should be synchronized with the PM choices of appropriate activities in their modes. The optimization process is responsible for finding concrete architecture alternatives that conform to the topology, driven also by a predefined set of system or subsystem attributes called *design objectives*, and constraints grouped by different types of analysis, called *analysis viewpoints*. Analysis viewpoints help calculate the system variables or define their feasible region for architectural optimization, considering concerns such as cost, weight, reliability, timing, resource allocation, and power and data distribution. The second preliminary work [17] defines the concept of *pluggable analysis viewpoints*, demonstrating its ability to specify design objectives as a library of reusable assets. The pluggable viewpoints are based on the concept of *classification-by-property* [18], in which the computational semantics for the sets appearing in constraints is specified according to different properties of the domain elements. Masin et al. [17] demonstrate that applying classification-by-property principles creates robust analysis libraries that remain resilient to system evolution during product design, and enables Lego type interoperability between different system analyses. An adaptation of AO to PM is straightforward using the methodology for pluggable analysis viewpoints: the AO-PM formulation below just adds “analysis viewpoint” AO-MRCPSP to other viewpoints, where constraints (25) are similar to (21) applied to subsystem/component catalogs.

$$\text{AO} - \text{PM} \quad \text{Minimize } \textit{Original objectives, Objectives}(1) \quad (23)$$

$$\text{Subject to}$$

$$\textit{Original architectural constraints} \quad (24)$$

$$\text{Constraints}(2)-(22)$$

$$\textit{Subsystem/component type synchronization constraints} \quad (25)$$

2.3 EMI Methodology

EMI methodology focuses on the integration of System Architecture Synthesis with Project Time Management. In most MBSE and PM methodologies, both tasks are relatively independent, performed by different people using different tools. EMI defines what information is transferred between the tools to implement the AO-PM model and obtain a holistic system architecture and project plan.

The initial information comes from the SE team to define system-related data such as subsystems, subsystem options, and performance goals. Then, the PM team defines project activities including precedence between activities, alternative modes, and required resources. The PM data is incorporated with other viewpoints considered during the design space exploration (DSE) for MCDM. The selected Pareto optimal solutions are transferred to the PM team for detailed analysis. If the results are satisfactory, the project can start. Otherwise, the PM team should update project parameters, especially the adjusted resources capacities, activity and resource buffers, and mode durations. New architectures are then found by the AO-PM model. The whole process is shown in Fig. 1. EMI can be used as a pre-project fuzzy front-end stage [19] in the early planning stages of the project or during the project. Each usage requires slight modifications in the suggested

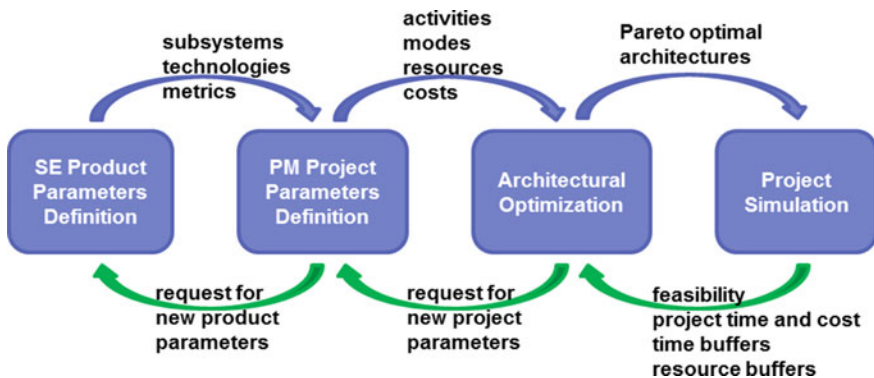


Fig. 1 EMI process

process, for example, customer interaction in fuzzy front-end or re-iteration to address system and environment changes during the project. Although we distinguish between SE and PM, in practice, the teams could be interdisciplinary.

3 Demonstration of EMI Methodology

3.1 *SE and PM Tools*

In this section we briefly describe two state-of-the-art tools for SE and PM that we use to demonstrate the EMI methodology. System engineers are typically responsible for creating alternative architectural solutions according to all requirements and goals, and for choosing the best one. However, the ever-increasing complexity of systems, strict design constraints, conflicting goals, and many other factors make the process of finding optimal designs extremely difficult. The common means to achieve system goals is to build optimization models for the set of specified architectures and find the best one using suitable optimization software tools. Unfortunately, this approach is highly labor-intensive because each architecture solution created by the engineer requires a separate optimization model created by an expert. This issue is resolved by the **Architectural Optimization Workbench (AOW)** described in [15, 17], and [20]. In the AOW, the system engineer can rapidly create the necessary system architecture, satisfying all functional and technical constraints needed to achieve the specified goals. Using standard SysML with the *concise profile* described above, the system engineer can model the composition rules (also known as architectural patterns, or templates) of the required functional and physical system structures and relations inside (data flow, energy flow, etc.) and between them (potential mapping between functions and physical components). In Rational Rhapsody [21] this approach immediately allows linking the functional models to the requirements in DOORS. The potential physical components are imported from a library, along with geometrical data, if relevant for the use case. Currently, AOW is integrated with MS Excel [22] and Pacelab Suite [23]. The optimization goals are specified as SysML constraints or Parametric Diagrams [20]. The tool uses all the inputs above to automatically generate a mathematical optimization program in OPL language [24] and the IBM Cplex solver. The AOW can be extended to produce optimization models in other languages, such as AMPL [25], to use with other solvers. Since there are multiple and usually conflicting goals, the optimization finds diverse Pareto optimal solutions (solutions where no goal can be improved without adversely affecting another). This is the maximum that can be done automatically before the final human decision. The results of the optimization are back-annotated into the SysML tool for the engineer to review. The AOW interface enables: importing and editing of the data, adding constraints and objectives, and managing the optimization runs, including viewing the results, and exporting them to the follow-on processes.

Our second tool, the **Project Team Builder** (PTB) is an integrated decision support system designed to support new product development teams during the project [26–32]. PTB combines simulation and case study approaches. Each case study is a new product development project performed under schedule, budget, and resource constraints, in a dynamic stochastic environment. The details of these case studies are built into the simulation, and all the data required for analysis and decision-making is easily accessed by the user interface. Random effects simulate the uncertainty in the environment, and decisions made by the user cause changes in the state of the simulated system. PTB supports a model-based approach for translating the voice of the customer into the project scope. A database is built into the PTB to support decision-making, post factum analysis, and backtracking. A friendly GUI enables a typical user to learn how to use the PTB within an hour. The PTB combines classical PM domains such as scheduling of activities with management of requirements. It offers a module for managing system requirements that supports the process of selecting alternative designs to determine system performance. The simulator allows the generation of project scenarios that include stochastic activity duration, resource capacity, and costs. Based on the input, the simulator offers a forecast for the project cost, schedule, and the product quality.

3.2 *EMI Use Case with Doors Management System*

To evaluate the benefits of our EMI methodology, we applied it to a use case involving the Doors Management System (DMS) for an aircraft. The DMS controls the latching and locking of doors in an aircraft. It communicates with the aircraft's pressurized system and consists of sensors, actuators, controllers, data collectors, and an Avionics Full-Duplex Switched Ethernet (AFDX) network. Functional requirements define the system functions, such as sensing, latching, locking and controlling, and the data flow between them. The structure describes potential DMS topologies, and the geometry gives potential physical locations. Figure 2 shows the transition of a customer story to a *concise model* in which all requirements and design alternatives are formally defined. The system optimization criteria include system weight, cost, and power consumption. Marketing requires project completion within one year (48 weeks).

As a comparison to EMI, we applied the SE methodology to the design of DMS for commercial aircraft. The AOW takes into account the system design for the aircraft door sensors, remote data concentrators, controllers, actuators and related power cables, power and data distribution flow, and safety requirements while optimizing the results for weight, cost, and power consumption. The AOW found four Pareto optimal solutions, shown in Fig. 3a. Systems engineers prefer to focus on two solutions with weight less than 200 kg, and power less than 305 W.

We also applied the PM methodology to the use case using PTB. The project activity network for the DMS use case is shown in Fig. 3b. Project simulations run in PTB showed that none of the two architectures coming from the SE team could

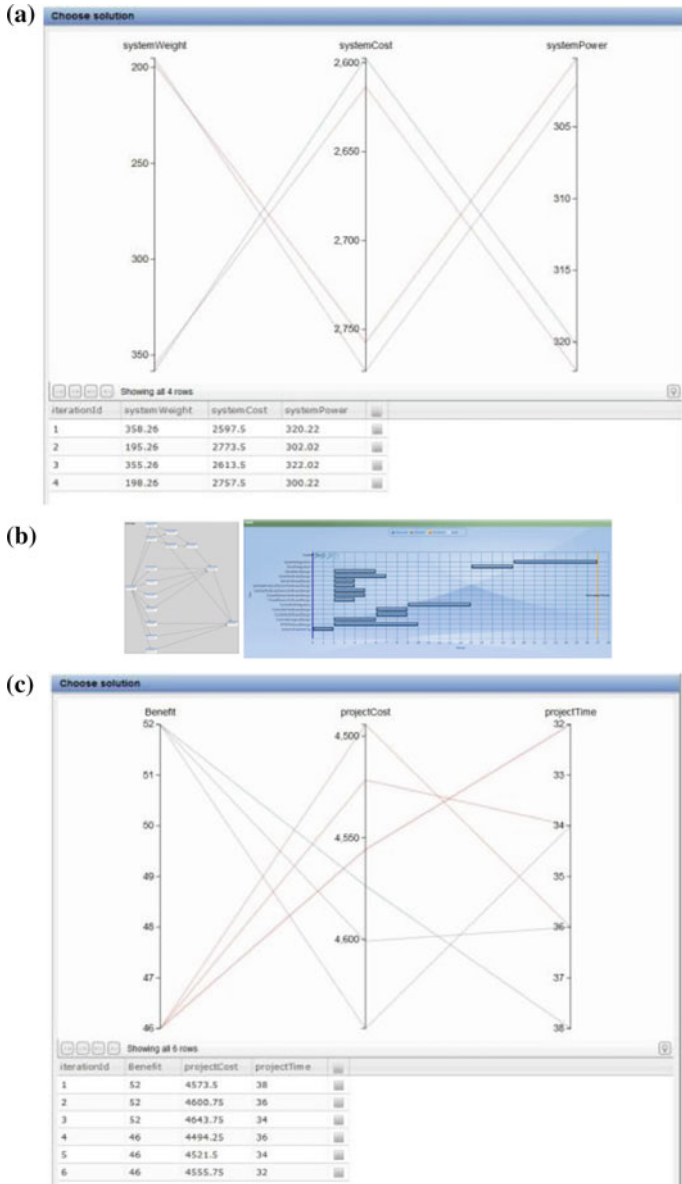


Fig. 3 a AOW only Pareto frontier. b DMS development activity precedence diagram and Gantt chart. c PTB only Pareto frontier

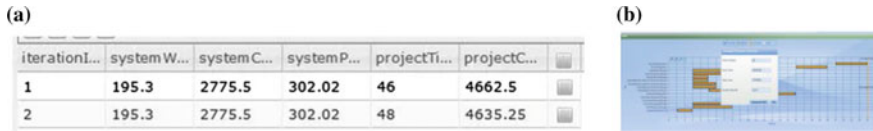


Fig. 4 a AOW AO-PM Pareto frontier. b PTB Gantt and simulation the first design

4 Summary

The impact of systems engineering on program cost was recognized over a decade ago [33], suggesting that from the very early stages, careful management of the relationships between the product and the project is crucial to the success of any project that aims to deliver a defined product. Systems engineers are required, therefore, to apply science and technology, as well as technical planning, management, and leadership activities [34]. While the technical issues are related to the product domain, the managerial aspects reside in the project domain.

In this paper, we presented a new approach, called EMI, to integrate SE and PM methodologies. EMI defines what information is transferred between the tools to implement the AO-PM model and obtain a holistic system architecture and project plan. Our work includes the mathematical foundation for EMI, implementation in AOW and PTB tools, and a detailed use case of DMS development for commercial aircraft. EMI can be used in pre-project fuzzy front-end stage [19], in early planning stages of the project, or during the project.

For future research we suggest handling project uncertainty in activity durations and resource availability by robust optimization, to reduce the number of iterations between AO and PM tools.

References

1. DAU, Systems Engineering, *Defense Acquisition Guidebook*, Chapter 4, Defense Acquisition University, 2006, http://akss.dau.mil/dag/GuideBook/PDFs/Chapter_4.pdf
2. Sage, A.P., Rouse, W.B. (eds.): *Handbook of Systems Engineering and Management*, 2nd edn. Wiley, New York (2009)
3. Forsberg, K., Mooz, H.: The relationship of system engineering to the project cycle. In: NCOSE and ASEM (1991)
4. Gulati, R.K., Eppinger, S.D.: The coupling of product architecture and organizational structure decisions. In: Working Paper. MIT (1996)
5. Sosa, M.E., Eppinger, S.D., Rowles, C.M.: Identifying modular and integrative systems and their impact on design team interactions. In: *Jof Mechanical Design* (2003)
6. Cataldo, M., Herbsleb, J.D., Carley, K.M.: Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. In: 2nd ACM-IEEE International Symposium on ESEM (2008)
7. MacCormack, A., Baldwin, C., Rusnak, J.: Exploring the duality between product and organizational architectures. *Res. Policy* (2012)

8. Avnet, M.S.: Information flow, team coordination, and shared knowledge in integrated concurrent engineering. In: INCOSE (2014)
9. Project Management Institute: *PMBOK® Guide*, 5th edn (2013)
10. Schwindt, C., Zimmermann, J. (eds.): *Handbook on Project Management and Scheduling*, Volume 1, Springer (2015)
11. Zapata, J.C., Hodge, B.M., Reklaitis, G.V.: The multimode resource constrained multiproject scheduling problem: alternative formulations. *AIChE J.* (2008)
12. Cplex Optimizer. www.ibm.com/software/commerce/optimization/cplex-optimizer/
13. Artigues, C.: Recent developments in mixed integer linear programming formulations for the resource-constrained project scheduling problem. *PMS* (2014)
14. Koné, O., Artigues, C., Lopez, P., Mongeau, M.: Event-based MILP models for resource-constrained project scheduling problems. In: *C&OR* (2011)
15. Broodney, H., Dotan, D., Greenberg, L., Masin, M.: Generic approach for systems design optimization in MBSE. In: *INCOSE* (2012)
16. OMG Systems Modeling Language. <http://www.omgsysml.org/>
17. Masin, M., Limonad, L., Sela, A., Boaz, D., Greenberg, L., Mashkif, N., Rinat, R.: Pluggable analysis viewpoints for design space exploration. In: *CSER* (2013)
18. Parsons, J., Wand, Y.: Emancipating instances from the tyranny of classes in information modeling. *ACM Trans. Database Syst.* **25**(2) (2000)
19. Katz G.: Rethinking the product development funnel (2011)
20. Masin, M., Broodney, H., Brown, C., Limonad, L., Mashkif, N., Sela, A.: Reusable derivation of operational metrics for architectural optimization. In: *CSER* (2014)
21. Rational® Rhapsody. <http://www-01.ibm.com/software/awdtools/rhapsody/>
22. MS Excel. <https://products.office.com/en-us/excel>
23. Pace Lab Suite. <http://www.pace.de/products/preliminary-design/pacelab-suite.html>
24. Van Hentenryck, P.: The OPL optimization programming language (1999)
25. AMPL. <http://www.ampl.com/>
26. Parush, A., Davidovitch, L., Shtub, A.: Simulation-based Learning in engineering education: performance and transfer in learning project management. *JoEE* (2006)
27. Davidovitch, L., Parush, A., Shtub, A.: Simulation-based learning: the learning-forgetting-relearning process and impact of learning history. *C&E* (2008)
28. Shtub, A., Parush, A., Hewett, T.: Guest editorial: the use of simulation in learning and teaching. *IJEE* (2009)
29. Parush, A., Davidovitch, L., Shtub, A.: Simulator-based team training to share re-sources in a matrix structure organization. *IEEE Trans. EM* (2010)
30. Shtub, A.: Simulation based training (SBT)—the next generation of project management training. *PM World J.* (2013)
31. Shtub, A., Iluz, M., Gersing, K., Oehman, J., Dubinsky, Y.: Implementation of lean engineering through simulation based training. *PM World J.* (2014)
32. Cohen, I., Iluz, M., Shtub, A.: A simulation-based approach in support of project management training for systems engineers. *Syst. Eng.* (2014)
33. Defense Systems Management College: *Systems Engineering Fundamentals* (1999)
34. Frank, M.: Cognitive and personality characteristics of successful systems engineers. *INCOSE* (2000)