

Project Management vs. Systems Engineering Management: A Practitioners' View on Integrating the Project and Product Domains

Amira Sharon,^{1,*} Olivier L. de Weck,² and Dov Dori¹

¹Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Haifa 32000, Israel

²Engineering Systems Division, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139

Received 3 August 2010; Revised 18 December 2010; Accepted 18 December 2010, after one or more revisions

Published online in Wiley Online Library (wileyonlinelibrary.com).

DOI 10.1002/sys.20187

ABSTRACT

While most Systems Engineering Management (SEM) applications use some subset of traditional Project Management (PM) methods and tools, the actual practice of systems engineering management involves continuous cognitive zigzagging between systems engineering—the product domain—and project management—the project domain. Focusing on seven PM methods, we examine two research questions regarding systems engineering practitioners: (1) While conducting SEM, do they perceive a notion of a project domain, a product domain, and a combined project-product domain? (2) What is the extent to which, and ways by which, systems engineering practitioners deem PM methods as effective for supporting SEM? Using analysis of structured questionnaires among 24 participants, we verified that project and product are indeed viewed as two complementary facets of SEM, and that certain PM methods address both domains better than others with respect to particular examined factors. © 2011 Wiley international, Inc. Syst Eng

Key words: systems engineering management; project management; model-based systems engineering; model-based project planning; project-product lifecycle management

1. INTRODUCTION

Systems Engineering (SE) and Project Management (PM) are two tightly intertwined domains as stated in the *Handbook of Systems Engineering and Management* by Sage and Rouse [2009]. The handbook is a recent detailed source for a framework to SEM and its application for various types of systems. Systems Engineering Management (SEM) utilizes some common project management framework features, such as Work

Breakdown Structure (WBS), project organization, and management plan in the form of System Engineering Management Plan (SEMP) [Blanchard, 2008]. As SEM is the practice that couples the SE domain and the PM domain, the successful implementation of system engineering requires not only technical but also managerial traits [Kossiakoff and Sweet, 2003]. Systems engineers are required therefore to apply science and technology, as well as technical planning, management, and leadership activities [Frank, 2000]. Systems engineering managers must rely on a combination of technical skills and management principles that address both complex technical and managerial issues. While the technical issues are related to the product domain, the managerial aspects reside in the project domain. Practicing SEM is a continuous process, conducted by iterative zigzagging between the technical product domain and the managerial project domain. It is an itera-

*Author to whom all correspondence should be addressed (e-mail: amirash@techunix.technion.ac.il; deweck@mit.edu; dori@ie.technion.ac.il).

tive process of planning, derivation, refinement, and simulation of both the product models and the management plans, while maintaining traceability and coherence between the product models and the project plan at all levels through the hierarchical structure of both the product and the project.

The impact of systems engineering on program cost was recognized over a decade ago [DSMC, 1999], suggesting that, from the very early stages, careful management of the relationships between the product and the project is crucial to the successes of a 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 and the product—for meeting the development and test objectives is a recipe for inadequate product performance and project time and cost overruns. The identification and management of the relationships between the two domains is at the heart of SEM practitioners. Empirical investigations have shown that the relationships and interactions between the architecture of systems, their development projects, and the organizational teams involved, should be aligned in order for a company to become successful [Eppinger and Salminen, 2001].

These observations call for a fresh look at the methods and tools systems engineering managers use in an attempt to bring the product issues and the project issues together under the same conceptual model of a combined project-product system ensemble. This is the premise of the Project-Product Lifecycle Management (PPLM) paradigm [Sharon, Perelman, and Dori, 2008; Sharon, Dori, and de Weck, 2009], the context within which this research was conducted. The PPLM approach is designed specifically with SEM in mind. The PPLM effort aims at developing a methodology and a support software environment for fusing the *product* to be developed with the *project* to be executed by SEM practitioners.

Focusing on seven researched PM methods, this paper presents two research questions: (1) While engaged in system engineering management, do practitioners perceive a notion of a project-domain, a product-domain, and a combined project-product domain? (2) How do systems engineers perceive the extent to which each one of the surveyed PM methods supports SEM?

2. THE COMPARED PROJECT MANAGEMENT METHODS

The need to concurrently address more than one domain has been identified by Suh [1990], who developed the Axiomatic Design (AD) methodology. AD addresses four domains: the customer, functional, physical, and process domains. The output of each domain evolves gradually and hierarchically from an abstract concept into detailed information. The hierarchical decomposition in one domain cannot be performed independently of the other domains; i.e., decomposition requires zigzagging in order to map adjacent domains to each other. The planning process involves the continuous processing of information within and between the four domains. The customer needs are identified in the customer domain and are stated in the form of required functionality of a product in the functional domain. Design parameters that satisfy the functional requirements are defined in the physical domain, and

manufacturing variables are defined in the process domain in order to determine how the product is to be produced. The AD physical domain is somewhat analogous to our product domain, while the AD process domain contains elements of our project domain. However, all the four AD domains focus to the product. In our context of SEM and PM, the mental zigzagging between domains entails switching back and forth between the product domain and the project domain. With this realization in mind, we have decided to conduct a comparison survey of Project Management Methods, which are commonly used by systems engineering managers. These methods include Work Breakdown Structure (WBS) [Department of Defense, 1968], Product Breakdown Structure (PBS) [Office of Government Commerce, 2005], Reliability, Availability, Maintainability (RAM) [Department of Defense, 2009], Critical Chain Project Management (CCPM) by Theory of Constraints (TOC), resource scheduling, procurement techniques, contract bidding techniques, and risk management methods. Other traditional methods are Earned Value Method (EVM), Design Structure Matrix (DSM), System Dynamics (SD), Critical Path Method (CPM), Program Evaluation and Reviewing Technique (PERT), and Gantt chart. The last methods, together with WBS, vary in terms of their objectives and applications, and are the most referenced PM methods in systems engineering handbooks sources, indicating that they are most commonly used within SEM. We have examined seven PM methods—EVM, DSM, SD, CPM, PERT, Gantt chart, along with Object Process Methodology (OPM) [Dori, 2002]—for answering our both research questions. See the Appendix for succinct description of the seven PM methods.

The potential use of Object Process Methodology (OPM) for project planning and management has been studied in the context of the Project-Product Lifecycle Management (PPLM) framework [Sharon, Perelman, and Dori, 2008, Sharon, Dori, and de Weck, 2009], and is explored further in this study. Originally used for modeling information systems and for systems architecting and engineering, Object Process Methodology (OPM) has been found useful for project planning in combination with product systems representation. Its suitability stems from its inherent capability to combine product- and project-based planning within a single Model-Based Project Plan (MBPP). OPM is a formal yet intuitive conceptual modeling language with formal ontology, syntax, and semantics. Specific extensions to OPM were defined as part of the PPLM research. Based on OPM's formality, specific PPLM OPM-based templates were developed to be used for modeling of combined project-product plans.

3. RESEARCH POPULATION AND SETTING

The research population consisted of 24 mid-career systems engineers from companies across the USA with an average of 8.4 years of work experience. The participants were from different companies, covering a wide range of industry sectors, including defense, telecommunication services, automotive, medical systems, and electronic design automation. The participants were among about 80 graduate students in the Systems Project Management course which is one of three core mandatory courses in the Systems Design and Management (SDM) program at MIT's Engineering Systems Divi-

sion. During the spring 2008 semester course, the research participants studied the seven project management methods surveyed in this study and practiced them through targeted homework assignments, along with specific number of 3-hour class sessions devoted to each method. Gantt and Object Process Methodology (OPM) were not explicitly taught as part of the course, since all participants were expected to have prior knowledge of these methods. Gantt was assumed as basic knowledge, and OPM was taught in a previous systems architecture course. Nevertheless, a short reminder of Gantt and OPM in the course was followed by introduction to Project-Product Lifecycle Management (PPLM) and Project-Product Lifecycle Management (MBPP), which spanned 1.5 sessions (4.5 h). The research was conducted through the fifth homework (HW5) given in the course, which was nonmandatory. The 24 respondents elected to do HW5 and participate in the study. Some of their motivation was the option they were given of having their final grade based on the best five out of six homework assignments.

An Unmanned Aerial Vehicle (UAV) development project case [de Weck and Lyneis, 2008] served as a running case study for all the homework assignments. The project concerns developing a UAV by a fictitious government-contracted leading UAV manufacturer.

For their first homework (HW1), all the participants were tasked with creating a simple SD model and exploring its behavior. They examined the impact of uncertainties in project assumptions on cost and schedule.

In HW2, they created a project plan using Critical Path Method (CPM), drew a project graph, estimated the early finish (EF) time of the project, and identified the critical path and slack times. Using Program Evaluation and Reviewing Technique (PERT), they had to analyze the impact of changes in individual task times on the critical path and consider probability distributions of task times and their effect on the project schedule.

HW3 called for applying Design Structure Matrix (DSM). Participants first translated the project graph from the previous assignment to a DSM representation. Next, they added iterations to the project and analyzed their effect on the previous task sequence. They then had to consider partitioning the DSM to reveal meta-tasks. Finally, they estimated the effect of these changes on the critical path and estimated project completion time.

For HW4, the participants focused on tracking projects and computing the various metrics defined in Earned Value Method (EVM) terms of cost and schedule in order to assess the overall performance of the project and to critically analyze and interpret the results.

Finally, based strictly on the text given in a previous homework assignment (HW2), HW5 called for creating two project plan versions, one using a Gantt chart model and the other using Object Process Methodology (OPM). In the second part of the homework the participants were asked to compare the seven project management methods they had studied in the course with respect to a set of 14 project management factors, as described in the next section.

4. RESEARCH METHODOLOGY

Since the investigated project management methods were taught in the course during lectures and practiced through homework assignments, we assumed that the participants had an identical minimal level of working knowledge of these methods. Since they were all practicing system engineers, the actual level of most of them was likely higher than this minimal assumed level.

The same system project case—the Unmanned Aerial Vehicle (UAV)—served as the basis for all the assignments throughout the entire course (except for the final projects). At each assignment the participants had to deal with another aspect of the same case. Had the experience the participants gained in applying all seven methods been drawn from different system projects, the influence of changing cases would not be separable from the applied methods..

4.1. The 14 SEM Factors and Their Dimensions

Recognizing that systems engineering management entails both the product and the project viewpoints, we defined 14 factors, F_1 – F_{14} , that account for both major classical project management issues and aspects of the joint project-product ensemble, which is at the focus of Systems Engineering Management (SEM).

The set of 14 factors was determined using an evaluation survey among practicing systems engineers, who were asked to select the most suitable factors from a larger amount of candidate factors. The extent of relevance of each one of the candidate factors to SEM was ranked by 20 practicing systems engineers, with average of 13.7 years of work experience, employed in a large enterprise dealing with development of diverse large-scale systems in the defense industry. The practitioners were asked to grade the extent of the relevance of each factor for the *project* domain and for the *product* domain, using a semi-Likert scale [Likert, 1932] of 1–5, where 1 is very little, 2 is little, 3 is moderate, 4 is high, and 5 is very. Not applicable was denoted by 0. The practitioners were not instructed to relate either to a specific project or product or to a specific scale of projects or products. Therefore, their rankings present a subjective judgment of their perception of each factor's extent of relevance for each domain. The medians of the rankings were calculated and set a threshold of 75% for the selection of factors to be used in the next research step. Using medians and the threshold has yielded a list of the 14 factors that were most agreed upon as suitable among the first-stage participants. The medians obtained for the 14 factors that passed the threshold span from 3.88 to 4.75, as shown in Table II. No significant difference was found between the rankings of both domains—the project and the product. This might indicate that, in the environment of large-scale project-product ensembles of the type the practitioners are involved in, the project and the product are strongly intercorrelated, resulting in only a nonsignificant difference. The surveyed Unmanned Aerial Vehicle (UAV) case, presented in this paper, is also in the ballpark of large-scale ensembles, though it was not actually conducted by the participants in a real enterprise environment.

Table I lists the 14 factors and their meaning, together with each factor's latent dimension. The set of all 14 factors to-

Table I. The 14 Systems Engineering Management Factors

#	SEM Factor	Meaning	Latent Dimension
1	Budget/Schedule measurement/tracking	The extent of ability to measure or track project budget or schedule	Project
2	Budget/Schedule forecasting	The extent of ability to forecast project budget or schedule	Project
3	Inter-relationships (process & product)	The extent of ability to comprehend inter-relationships between the project process and the product	Project -Product
4	Resource management	The extent of ability to manage project resources; allocate or assign resources manageably	project
5	Stakeholders/agents tracking	The extent of ability to track different stakeholders and agents both of the project and of the product	Project -Product
6	Performance quality	The extent of ability to track the quality of the product performance	product
7	Product quality	The extent of ability to plan the product quality	product
8	Product planning	The extent of ability to plan the product itself (functionality, architecture)	product
9	Product measurement/tracking	The extent of ability to measure and track the progression of the product itself	product
10	Risk management	The extent of ability to manage both the product risks and the project risks	Project -Product
11	Iterations management	The extent of ability to manage iterations in the project	project
12	Information resolution level	The extent to which the relevant information, both of the product and of the project , is at satisfactory resolution	Project -Product
13	Ease of communication	The extent of ease of communicating the relevant information, both of the product and of the project	Project -Product
14	Change management	The extent of ability to manage change (track, understand the impact) both in the product and in the project	Project -Product

gether forms the SEM dimension, D_{SEM} , as defined in Eq. (1). Four of the 14 SEM factors belong to the “classical” project management domain and are indeed addressed by common project management methods; therefore, they are categorized in the project dimension, $D_{project}$, as defined in Eq. (2). Four other factors fit traditionally in the product domain; hence they are categorized in the product dimension, $D_{product}$, as defined in Eq. (3). The remaining six factors are common to the combined product-project domain, $D_{project-product}$, defined in Eq. (4), as they cannot be uniquely associated with either the product alone or the project alone. With respect to risk management, we have adopted NASA’s view of this issue as common to the project and the product domains [NASA, 1995]. This approach is founded on the premise that there are technical risks, which are mostly in the product domain, and managerial risks, which are mostly in the project domain. This is contrary to the approach of leading standards IEEE Standard 1220 [IEEE, 1994] and ANSI/EIA Standard 632 [ANSI/EIA, 1999], which view risk management primarily as a managerial issue and therefore relate to it as project domain issue. A fourth dimension, $D_{Combined_project-product}$, is also defined in Eq. (5) for analysis reasons—this is a combination of the factors of $D_{project}$ and $D_{product}$.

The classification of the 14 factors is as follows:

$$D_{SEM} = \{F_1, F_2, \dots, F_{14}\}, \quad (1)$$

$$D_{project} = \{F_1, F_2, F_4, F_{11}\}, \quad (2)$$

$$D_{product} = \{F_6, F_7, F_8, F_9\}, \quad (3)$$

$$D_{project-product} = \{F_3, F_5, F_{10}, F_{12}, F_{13}, F_{14}\}, \quad (4)$$

$$D_{Combined_project-product} = \{F_1, F_2, F_4, F_6, F_7, F_8, F_9, F_{11}\}. \quad (5)$$

4.2. The Survey Questions and Their Analysis Method

The 14 factors were introduced to the participants in the random order listed in Table II. The participants were not instructed in any way to think specifically of the considerations as related to “project,” “product,” or “project-product” dimensions. Our aim was to explore whether their unguided perceptions towards the different factors would reflect recognition of these factors as related to our three predefined latent dimensions of “project,” “product,” and “project-product.”

Table II. The 14 Factors Extent of Relevance for SEM

#	SEM Factor	Dimension	Ranked relevance to project domain		Ranked relevance to product domain	
			Median	Standard deviation	Median	Standard deviation
1	Budget/Schedule measurement/tracking	(2), (5)	3.50	1.27	4.63	0.38
2	Budget/Schedule forecasting	(2), (5)	3.50	1.13	4.25	0.79
3	Inter-relationships (process & product)	(4)	3.63	1.51	3.75	1.50
4	Resource management	(2), (5)	3.75	1.07	4.00	1.07
5	Stakeholders/agents tracking	(4)	3.75	0.90	3.50	1.27
6	Performance quality	(3), (5)	4.50	0.79	4.00	1.53
7	Product quality	(3), (5)	4.75	0.38	4.00	1.41
8	Product planning	(3), (5)	4.50	0.79	4.00	1.41
9	Product measurement/tracking	(3), (5)	4.50	0.53	4.13	1.07
10	Risk management	(4)	3.75	0.76	4.38	0.79
11	Iterations management	(2), (5)	3.63	1.25	4.00	0.69
12	Information resolution level	(4)	3.63	1.35	3.38	0.79
13	Ease of communication	(4)	3.88	1.07	4.13	0.38
14	Change management	(4)	4.29	0.82	4.14	0.98

The participants were instructed to rank each one of the 14 factors for each one of the seven systems engineering management methods using the same Likert scale [Likert, 1932] as before.

Since the participants were practicing systems engineers, their views of the project management tools tended to reflect the application of these methods in systems engineering management more than in project management. To examine the participants' views of each project management method with respect to each factor, we compared the responses for each one of 14 factors with respect to each one of the seven PM methods.

To determine whether our classification of the 14 factors into the three domains can be verified by the research participants' responses, we first analyzed the grades they had given for each factor and method combination. Using the Alpha Cronbach coefficient [Cronbach, 1951], we determined whether the domain-categorized factors can be considered to be a distinct dimension.

Using (6), the Alpha Cronbach coefficient α is calculated for estimating how well a set of variables measures a single 1-dimensional underlying construct. k is the number of items, σ_i^2 is the variance of the i th item, and σ_T^2 is the variance of the total score formed by summing all the items of the tested sample. The result obtained from (6) determines the internal consistency of items within a single test, indicating reliability. The reliability is in terms of the ratio between the true score variance of the "underlying construct" and the observed score variance of that unidimensional construct, where the construct is the hypothetical variable that is being measured [Hatcher, 1994]. α ranges in value from 0 to 1, when 0.70 is defined [Nunnally, 1978] as the cutoff value to be an acceptable reliability. It increases when the average intervariable correlation increases. Therefore, high values of α provide evidence that the variables included in its calculation measure the same underlying construct.

$$\alpha = \frac{k}{k-1} \left(1 - \sum_{i=1}^k \sigma_i^2 / \sigma_T^2 \right). \quad (6)$$

The sum of all the participants' Likert scale rankings for each factor was calculated, and the sum of all 14 factors for each PM method was taken as that method's score. The α values for each PM method were calculated from the Likert scale results for each group of factors defined for each domain, namely, $D_{project}$ as defined in (2), $D_{product}$ as defined in (3), $D_{project-product}$ as defined in (4), and $D_{Combined_project-product}$ as defined in (5).

5. RESULTS AND ANALYSIS

We present and discuss the comparison of results of participants' rankings of the seven surveyed PM methods with respect to the 14 project management factors listed in Table II.

5.1. Methods Comparison by Factors

Using (6), α was initially calculated for all 14 factors, as presented in the first row, D_{SEM} , of Table III. The results are higher than 0.70 for all but the Design Structure Matrix (DSM) method. Therefore, we can use the participants' rankings for all the 14 factors for the sake of comparison between the six PM methods, from which DSM is excluded. When excluding two factors from $D_{project-product}$ — F_{12} (Information Resolution Level) and F_3 (Interrelationships, process & product)—DSM exceeds an α value of 0.7. Excluding these two factors for all the seven PM methods, we are left with a set of 12 factors that can be reliably used for the comparison of all the seven PM methods.

Figure 1 represents by the dark bars the sum of scores of the 14 factors participants assigned for each method. Object Process Methodology (OPM) scored the maximum sum, 885 points. A cutoff value of 664 points, which is 75% of this

Table III. Cronbach's Alpha Results for Factors Combinations

Factors Combination		Project Management Method						
		SD	PERT	CPM	DSM	EVM	Gantt	OPM
D_{SEM} (1)	α (6)	.743	.793	.754	.640	.757	.760	.855
	Best Improved	-	-	-	.702 ^a	-	-	-
$D_{project}$ (2)	α (6)	.738	.700	.422	.744	.505	.511	.731
	Best Improved	-	-	.608 ^b	-	.512 ^b	.613 ^b	-
$D_{product}$ (3)	α (6)	.775	.472	.402	-.343 ^c	.414	.655	.746
	Best Improved	-	.486 ^c	.601 ^d	.725 ^d	.560 ^f	.678 ^g	-
$D_{project-product}$ (4)	α (6)	.251	.535	.624	.523	.723	.605	.706
	Best Improved	.435 ^h	-	.687 ^h	.579 ^h	-	.651 ^h	-
$D_{Combined_project-product}$ (5)	α (6)	.807	.655	.565	.703	.524	.451	.826
	Best Improved	-	.734 ^d	.690 ^d	-	-	.570 ⁱ	-

^a Improved by deletion of F_{12} – Information Resolution Level and F_3 – Inter-relationships (process & product)
^b Improved by deletion of F_{11} – Iterations Management
^c The value is negative due to a negative average covariance among items.
^d Improved by deletion of F_8 – Product Planning
^e Improved by deletion of F_7 – Product Quality
^f Improved by deletion of F_9 – Product measurement/tracking
^g Improved by deletion of F_6 – Performance Quality
^h Improved by deletion of F_5 – Stakeholders/ agents tracking
ⁱ Improved by deletion of F_2 – Budget/Schedule forecasting

maximum score, leaves us with three methods: Object Process Methodology (OPM), System Dynamics (SD), and Earned Value Method (EVM).

The light grey bars in Figure 1 represent the sums of rankings of the 12 factors (where F_3 and F_{12} are excluded). With these 12 factors, SD scored the maximum sum, 769 points. Assigning a cutoff of 577, which is 75% of this maximum, leaves four methods in the game: OPM, SD, EVM, and DSM. The sum of all the participants’ rankings for each factor was calculated, and the sum of the 12 factors for each PM method was taken as that method’s final score.

5.2. Dimensions Analysis

We present and discuss the results of α calculated for the four dimensions $D_{project}$, $D_{product}$, $D_{project-product}$, and $D_{Combined_project-product}$.

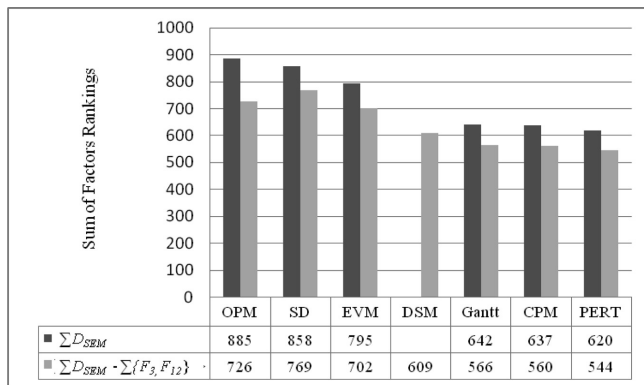


Figure 1. Project management methods comparison by sums of factors rankings.

5.2.1. The Project Dimension

Based on the participants’ rankings for the four factors of $D_{project}$, α was calculated for each PM method, and is presented in Table III. Since we evaluated *project* management methods, we expected α results with value of .70 or higher to be obtained for all seven methods, indicating that the factors in the underlying project domain are handled by all the seven PM methods. Surprisingly, however, as Table III shows, such above-the-cutoff values were found only for four PM methods: System Dynamics (SD), Program Evaluation and Reviewing Technique (PERT), Design Structure Matrix (DSM), and Object Process Methodology (OPM). The remaining three methods, namely, Critical Path Method (CPM), Earned Value Method (EVM), and Gantt, did not pass the 0.70 α cutoff acceptance value. Even for the best improved result, which was obtained by removing F_{11} —iterations management—these three methods still remain below the 0.70 cutoff value.

5.2.2. The Product Dimension

Applying a similar analysis for the product domain, we calculated α for the four factors of $D_{product}$ for each PM method. As Table III shows, the product dimension was found acceptable for three methods: System Dynamics (SD), Object Process Methodology (OPM), and Design Structure Matrix (DSM). The latter got in as having a product dimension only after removing F_8 —product planning.

5.2.3. The Project-Product Dimension

The project-product dimension was found to characterize only two methods: Earned Value Method (EVM) and Object Process Methodology (OPM) (see the third row in Table III). The results reflect the participants’ perception that only these two methods have an underlying project-product dimension.

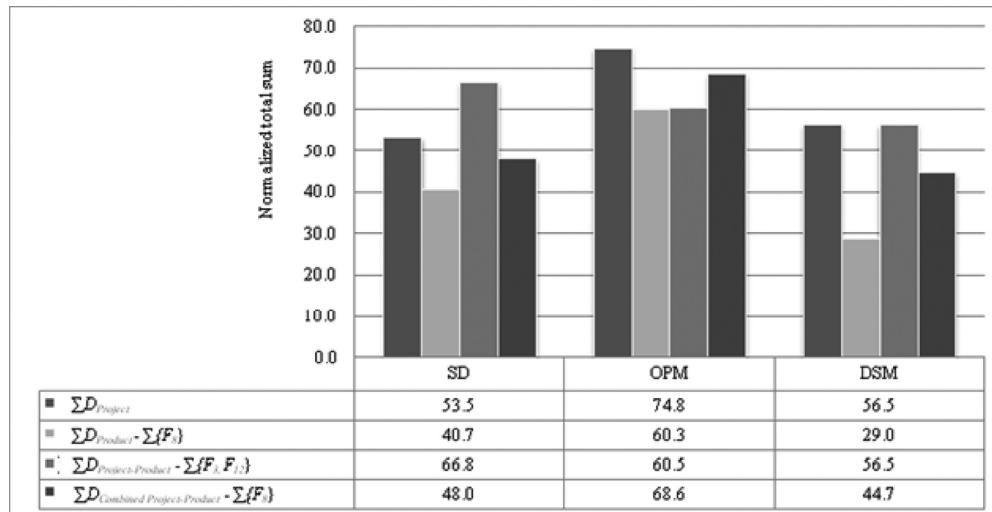


Figure 2. Project management methods comparison by dimensions.

In view of the small number of methods found for the project-product dimension, we also examined the combined project-product domain $D_{Combined_project-product}$. Although no such underlying dimension was predefined in the survey design, the participants’ rankings that yielded α value of 0.70 or higher might potentially reflect that the project and product dimensions are cognitively inseparable. This sits well with the results obtained in the gauging survey discussed earlier. As presented in the fourth row of Table III, the combined project-product dimension was found for Object Process Methodology (OPM), System Dynamics (SD), Design Structure Matrix (DSM), and Program Evaluation and Reviewing Technique (PERT). The latter became acceptable after elimination of F_8 —product planning—even though for the product dimension it had not reached the cutoff value.

5.3. Methods Comparison by Dimensions

Comparison of the seven PM methods by dimension can be conducted with the methods for which all dimensions were found—SD, DSM, and OPM. The comparison, presented in Figure 2, is based on calculated sums of scores participants assigned to each factor for each PM method, as listed in Table IV. Each relevant total sum S_{Total} was divided by the quantity of factors used for calculating that sum. The sum of all 14 factors, S_{Total} for ΣD_{SEM} , is not applicable for DSM since only by excluding F_{12} and F_3 did it pass the 0.70 threshold. The same rationale was followed when excluding factors for reliably calculating the sums for $\Sigma D_{project-product}$ and for $\Sigma D_{Combined_project-product}$. In addition, the sum of the product dimension factors, S_{Total} for $\Sigma D_{product}$, was calculated for all three methods without F_8 , since DSM got in as having the product dimension only after removing this factor.

Based on the data presented in Table IV, Figure 2 shows for each of the three methods in the table the sum of scores participants assigned to that method, for each one of the dimensions. The values on the vertical axis represent the normalized total sum, which is S_{Total} divided by the number of factors, for each PM method.

The project dimension scores (darkest grey bars, $\Sigma D_{project}$) are higher than the product dimension scores (lightest grey bars, $\Sigma D_{product} - \Sigma\{F_8\}$) for all the three PM methods. The scores of the combined project-product dimension (dark grey bars, $\Sigma D_{Combined_project-product} - \Sigma\{F_8\}$) are reasonably situated between the project and the product dimension scores. OPM scored the highest in three dimensions—project dimension, product dimension, and the combined project-product dimension. While for SD and DSM, the project-product dimension scores are higher than those of the combined project-product dimension, the result is reverse for OPM. This may indicate that the dimensions perception is more complex: for OPM, a project-product dimension is acceptable, but the separate project dimension and product dimension are not only accept-

Table IV. Sums Calculations for Comparison of Methods by Dimensions

Factors Combination	Sum	Project Management Method		
		OPM	SD	DSM
ΣD_{SEM}	S_{Total}	885	858	N/A
	$S_{Total}/14$	63.2	61.3	N/A
$\Sigma D_{SEM} - \Sigma\{F_3, F_{12}\}$	S_{Total}	726	769	609
	$S_{Total}/12$	60.5	64.1	50.8
$\Sigma D_{project}$	S_{Total}	214	299	226
	$S_{Total}/4$	53.5	74.8	56.5
$\Sigma D_{product} - \Sigma\{F_8\}$	S_{Total}	122	181	87
	$S_{Total}/3$	40.7	60.3	29.0
$\Sigma D_{project-product} - \Sigma\{F_3, F_{12}\}$	S_{Total}	267	242	226
	$S_{Total}/4$	66.8	60.5	56.5
$\Sigma D_{Combined_project-product} - \Sigma\{F_8\}$	S_{Total}	336	480	313
	$S_{Total}/7$	48.0	68.6	44.7

able, but also rank higher. For SD and DSM the perception of dimensions is reversed.

5.4. Discussion

Due to the diversity of the Systems Engineering Management (SEM) vocation and the wide range of practitioners' training and experience, it is very difficult, if at all possible, to find a group of systems engineers who are homogeneous in their knowledge and application of the traditional Project Management (PM) methods. On top of this, a group of systems engineers who are familiar with Object Process Methodology (OPM) in addition to the traditional PM methods had to be identified. This is hardly achievable since model-based project planning is still being developed, and OPM is not yet widely used for constructing a model-based project plan. Fortunately, the 24 midcareer systems engineers who studied in the SDM graduate program at MIT and constituted our research group meet these criteria. We encouraged the participants to reflect on their work and provide criticism, and so they did. Most participants noted that the Model-Based Project Plan (MBPP) approach seemed promising but was not yet mature to be usable in practical applications. Nevertheless, when the participants ranked the PM methods with respect to the given set of factors, their judgment reflected a favorable attitude towards MBPP. Since the participants were asked to record the duration of the modeling tasks using Gantt and OPM, their reflections indicated that they had thought we are comparing modeling durations rather than our more subtle research goals. Given these, we can exclude potential bias in favor of MBPP.

Although the participants were not instructed to group the factors in any specific way, they cognitively combined the factors in a way that three out of the seven examined methods—System Dynamics (SD), Design Structure Matrix (DSM), and Object Process Methodology (OPM)—were found to handle well both the project dimension and the product dimension. This outcome is different than the one obtained in the gauging survey among 20 practitioners, where no significant difference was found between the domains. This might indicate that when no specific project or product is concerned—as was the case in the gauging survey—systems engineers tend to strongly intercorrelated the project with the project in their minds, influenced by a “general” SEM practicing environment. The separation of domains emerges and becomes significant in the context of a specific project-product ensemble, which in our study was the Unmanned Aerial Vehicle (UAV) case.

Object Process Methodology (OPM) was found to be the only method suitable for the project, product, and project-product dimensions. This outcome supports the anticipated role of OPM as a “common paradigm and language” [INCOSE, 2004] for communication among stakeholders and management of multidisciplinary teams of experts who are partners in the systems engineering management process. Since OPM had been selected as the modeling basis for the Project-Product Lifecycle Management (PPLM) framework, this is an encouraging finding.

6. CONCLUSION

The research presented in this paper is based on a specific simplified Unmanned Aerial Vehicle (UAV) case. The findings support the notion that the project and the product are two complementary facets of systems engineering management. For the UAV case, with the specific population of 24 midcareer systems engineers, the results indicate that, for the purpose of Systems Engineering Management (SEM), System Dynamics (SD), Design Structure Matrix (DSM), Earned Value Method (EVM), and Object Process Methodology (OPM) are likely to be more useful than Program Evaluation and Reviewing Technique (PERT), Critical Path Method (CPM), and Gantt. While management practices vary widely across demographics, in this study, based on participants with predominantly US experience, OPM was found the most suitable method both by dimensions analysis and ranking comparison analysis. This result indicates that OPM should be favorably considered as a suitable basis for managing product-project ensembles within systems engineering management. Applying the Project-Product Lifecycle Management (PPLM) methodology, it might become the actual bridge between systems engineering and project management, enabling simultaneous expression of the function, structure and behavior of both the project and the product within a holistic integrated conceptual model. OPM is currently in the process of becoming an ISO standard for enterprise standards. When completed, this endorsement will enable accelerated dissemination of Object Process Methodology (OPM) as a basis for enterprise standards in general and for Project-Product Lifecycle Management (PPLM) in particular.

Although the subjects in the research are practicing systems engineers, the research was conducted in an academic environment. Therefore, as future work, we propose that this research be conducted in an industrial environment. Such a research could also examine the implications on a real large-scale project, as opposed to the simplified UAV case reported in this paper.

APPENDIX: THE COMPARED PROJECT MANAGEMENT METHODS

The seven PM methods discussed in this paper vary in terms of their objectives and applications, as listed in Table V. Following are separate subsections succinctly describing the seven methods.

A.1. The Critical Path Method and Program Evaluation and Reviewing Technique

The *Critical Path Method* (CPM) is a network model of the project, developed in the 1950s by DuPont for management of maintenance projects in industrial factories. CPM depicts tasks along with dependency information, duration, and the slack time for each activity. CPM chart time is deterministic, resulting in a fixed estimate of the time required to complete the project. The method shows the activities that are critical to maintaining the schedule. Delay in an activity on the critical path delays the entire project. To accelerate the project, it is

Table V. The Seven Investigated Project Management Methods

PM Method	System Dynamics	Program Evaluation and Reviewing Technique	Critical Path Method	Design Structure Matrix	Earned Value Method	Gantt chart	Object Process Methodology
Short Name	SD	PERT	CPM	DSM	EVM	Gantt	OPM
Main Objective	Project planning – Dynamic Modeling	Project planning: Scheduling	Project planning: Scheduling	Project planning, Product design	Project control	Project planning: Scheduling	Combined project planning and product modeling & design

necessary to reduce the total time required for the activities performed in sequence along the critical path. Figure 3 depicts an example CPM chart of the Unmanned Aerial Vehicle (UAV) case discussed in this paper with its 23 tasks, their slacks, and the relationships. The three critical paths are denoted by solid lines.

Like CPM, the *Program Evaluation and Reviewing Technique* (PERT) is a network model, developed in the late 1950s as part of the U.S. Navy’s Polaris project. Similar to the CPM chart, PERT depicts tasks along with dependency information and duration. This allows for assigning parametric probabilities to task completion times in accordance with optimistic, pessimistic, and likely estimations. Although both CPM

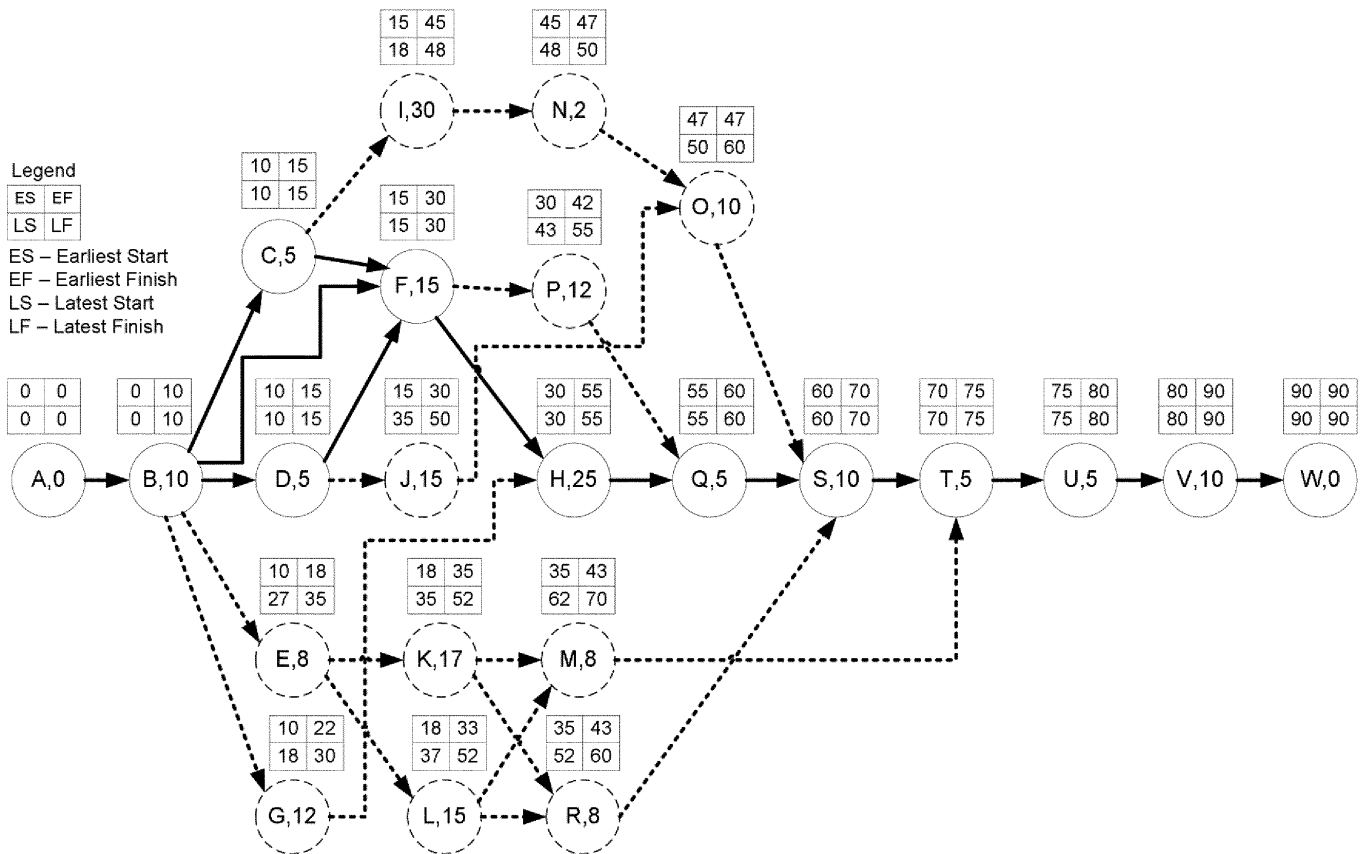


Figure 3. Example CPM chart of the Unmanned Aerial Vehicle (UAV) case.

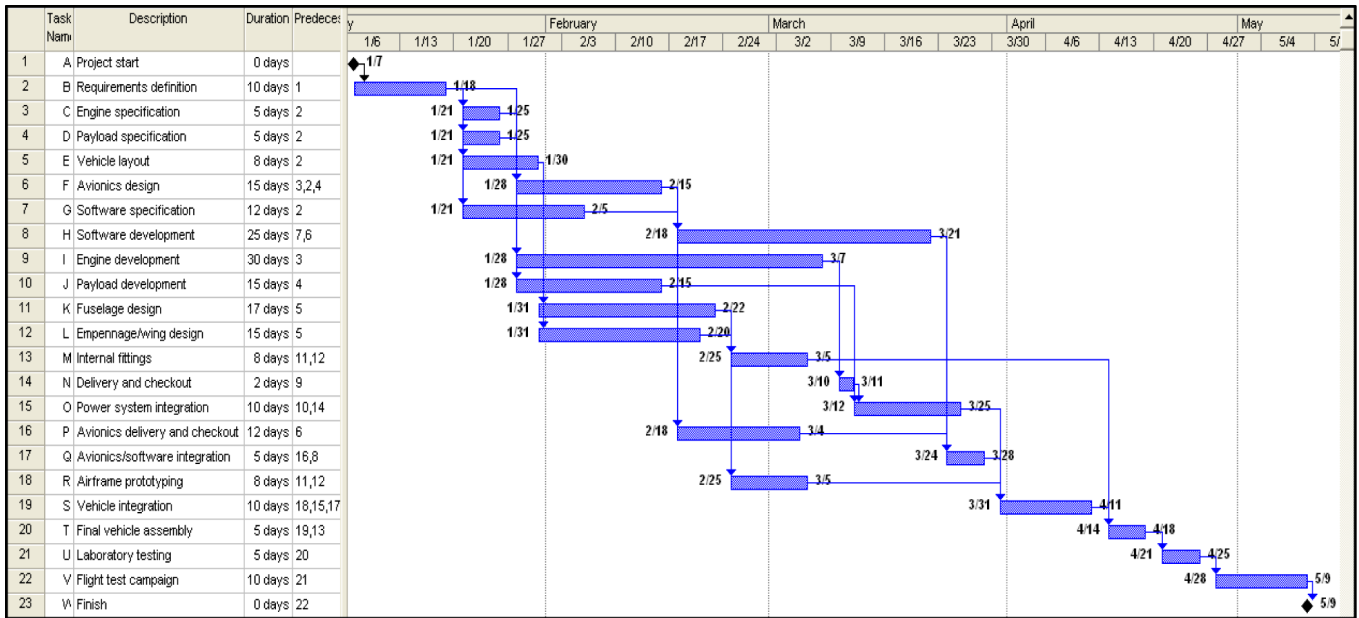


Figure 4. Gantt chart of the UAV case. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

and PERT models contain all the information, it is difficult to track the project progress using these network diagrams.

A.2. The Gantt Chart

The *Gantt chart*, shown in Figure 4 for the same UAV case, is probably the most widely used method for project planning. The Gantt chart comprises horizontal scheduling bars with time flowing from left to right, allowing for both planning and tracking of project schedule. In its original form, invented roughly forty years before PERT and CPM, the Gantt chart showed the timing of tasks without specifying relationships among them. Constraining relationships between tasks, such as start-to-start and finish-to-start, shown as the vertical directed lines in Figure 4, were added only in the late 1990s, making it possible to view and understand the impact of a single task delay on the entire project duration. Indeed, in its latest form, Gantt charts are used also for CPM-based project analysis, using software tools to extend the network models representation.

A.3. The System Dynamics Method

System Dynamics (SD) was developed initially from the work of Jay W. Forrester [Forrester, 1961, 1973]. Over the last 20 years, SD has been used to model complex development projects in order to improve their performance [Lyneis and Ford, 2007]. System dynamics models are created and used for project planning, addressing planned budget, schedule, resources, risks, and past experience. SD can help map out iteration issues and highlight rework loops. During the project planning process, SD models can predict effects of changes for different scenarios on a project’s productivity and quality due to various factors. Figure 5 is an SD model of the UAV

case depicted in both the PERT network in Figure 3 and the Gantt chart in Figure 4. In the CPM and Gantt models, the 23 tasks are shown with varying amounts of effort, while, in an aggregate SD model, all the tasks should be of approximately the same amount of effort. Therefore, the model of the project consists of approximately 100 equivalent tasks (the exact number does not matter, as the productivity would change correspondingly). Figure 5 depicts the graph of amount of work (number of tasks out of 100) done as a function of time for two staffing quality levels: perfect quality (1.0) and quality of 0.7.

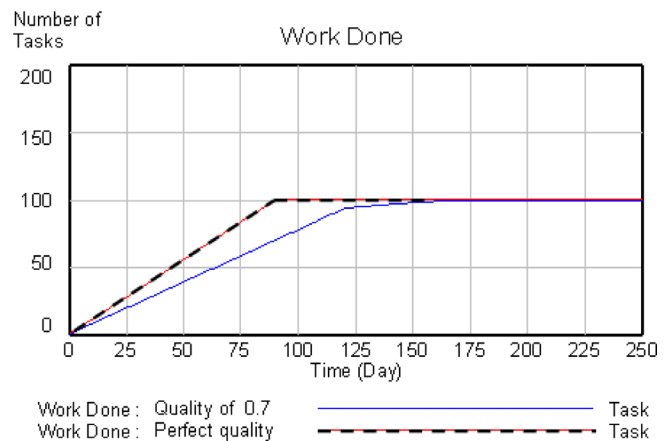


Figure 5. System Dynamics chart of the UAV case [Lyneis and Ford, 2007]. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	
a	a																							
b	X	b																						
c		X	c																					
d		X		d																				
e		X			e																			
f		X	X	X		f																		
g		X					g																	
h						X	X	h																
i			X						i															
j				X						j														
k					X						k													
l					X							l												
m									X	X			m											
n									X					n										
o										X					o									
p					X											p								
q							X									X	q							
r										X	X							r						
s												X			X	X	X	s						
t												X						X	t					
u																			X	u				
v																				X	v			
w																					X	w		

Figure 6. Example DSM [Lyneis and Ford, 2007]. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

A.4. The Design Structure Matrix

The *Design Structure Matrix* (DSM) is a square matrix representation of interactions among entities in a system. In project management, it enables modeling and analyzing dependencies among components, tasks, or teams in a project. The use of matrices in system modeling can be traced back to Warfield in the 1970s and Steward [Steward, 1981]. In the 1990s, the method received attention and became widespread. Eppinger et al. [1994] used a matrix representation to capture both the sequence of and the technical relationships among design tasks, which were then analyzed in order to find

alternative sequences and/or definitions of the tasks. Eppinger used DSM also in the context of project management.

Browning [2001] suggested four different types of DSM according to the data that the DSM represents: (1) component-based DSM, which represents relations among objects in the system, for use in systems engineering and architecting; (2) task-based DSM, which represents relations among tasks or activities—processes in the system, for use in project management; (3) parameter-based DSM, which is similar to (2) but at a lower level; and (4) team-based DSM, which interfaces between teams, for use in organizational design and multiteam projects. The task-based DSM provided in Figure 6 shows the same 23 tasks and relationships of the UAV case after matrix manipulations that avoid iterations, since all the interactions are below the matrix diagonal.

A.5. The Earned Value Management

Used for project performance measurement, the *Earned Value Management* (EVM) is a control method based on conversion of scope of work to budget terms. The underlying concept of EVM has been in use for over a century, and the US department of defense has used EVM in a modest measure back in the 1960s. EVM was elaborated in recent years and became an ANSI standard, which provides a forecast to a project’s end along with indications of exceptions to the plan. The basic idea behind EVM is comparing planned and actual work in order to determine budget and schedule propagation. The earned value can only be calculated after a project’s actual start, and it is always defined in reference to a specific control point in time. The different EVM parameters are calculated at the task level and are summed up in an aggregate manner for the entire project. Figure 7 is an example of an EVM plot, depicting for the same project shown in Figures 3, 4, and 6, the Curves of Planned Value (PV), Actual Cost (AC), and Earned Value (EV) against each other from the beginning of the project till the “Data Date” time point.

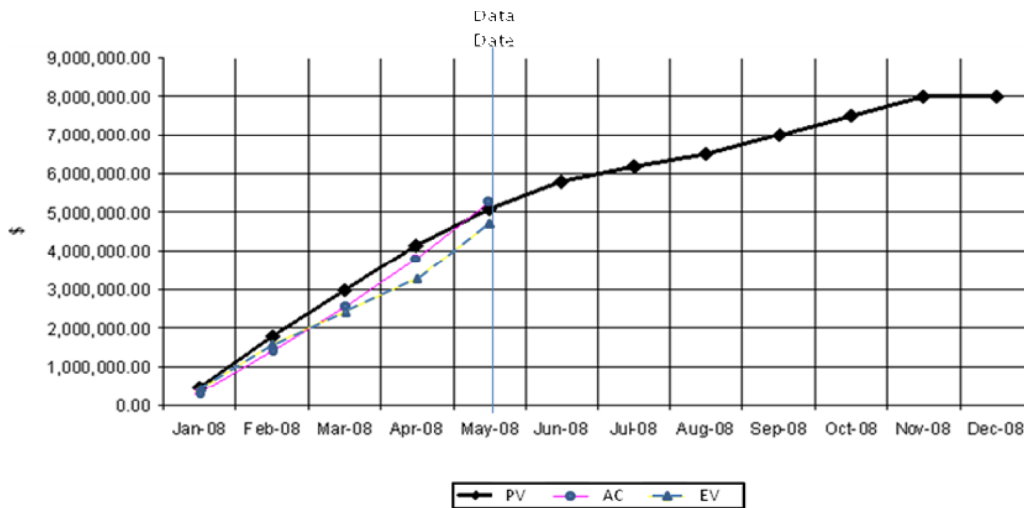


Figure 7. EVM Chart of the UAV project [INCOSE, 2004]. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

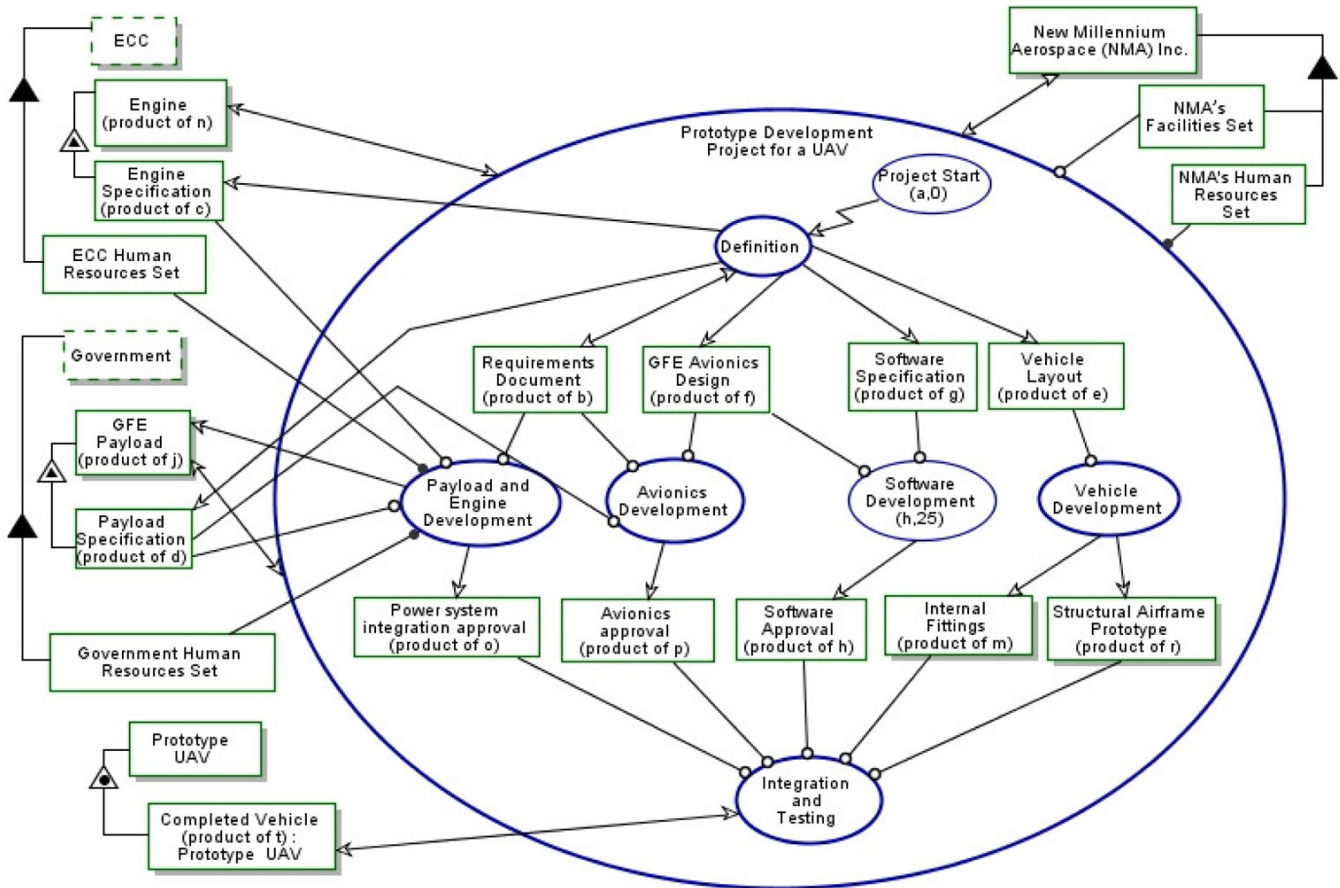


Figure 8. Object-Process Diagram (OPD) of the UAV case. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

A.6. Combined Project-Project Model-Based Plan

The Combined Project-Project Model-Based Plan is a derivative of the Project-Product Lifecycle Management (PPLM) approach [Sharon, Perelman, and Dori, 2008; Sharon, Dori, and de Weck, 2009], which integrates the project domain with the product domain via a shared ontology that explicitly relates project entities to product entities within a unifying frame of reference, utilizing Object Process Methodology (OPM) [Forrester, 1973] as the underlying conceptual modeling paradigm and language.

OPM is a formal yet intuitive paradigm for systems architecting, engineering, development, lifecycle support, and evolution. It has been used for modeling natural and artificial complex systems, where artificial ones might comprise humans, physical objects, hardware, software, regulations, and information. As its name suggests, the two basic building blocks in OPM are (stateful) objects—things that exist (at some state), and processes—things that transform objects by creating or destroying them, or by changing their state.

Object-Process Diagram (OPD) is the graphic representation of an OPM model, which has also an equivalent textual representation. Figure 8 is an OPD at the second level of detail from top of the same UAV case shown in the previous

figures. Tasks are modeled as processes, denoted by ellipses, while deliverables are objects, modeled by rectangles, which can include specifications, drawings, approvals, reports and other document types, prototypes, simulation and analysis results, as well as the final product. Structural links include whole-part (the black triangle) connecting a whole to its part(s), and characterization (black-on-white triangle), connecting an object with its attribute(s).

REFERENCES

ANSI/EIA, Processes for engineering a system, ANSI/EIA Standard 632, American National Standards Institute, Washington, DC, 1999.

B.S. Blanchard, System engineering management, Wiley, Hoboken, NJ, 2008.

T. Browning, Applying the design structure matrix to system decomposition and integration problems: A review and new directions, IEEE Trans Eng Management 48(3) (2001), 292–306.

L.J. Cronbach, Coefficient alpha and the internal structure of tests, Psychometrika 16 (1951), 297–334.

Department of Defense (DoD), Work breakdown structures for defense material items, Military Standard Mil-Std-881, Washington, DC, 1968.

- Department of Defense (DoD), Reliability, availability, maintainability, and cost rationale report manual, Office of the Secretary of Defense, Washington, DC, 2009.
- O. de Weck and J.M. Lyneis, MIT ESD.36 System Project Management course assignments, Massachusetts Institute of Technology, Cambridge, MA, 2008.
- D. Dori, Object-process methodology—a holistic systems paradigm, Springer, New York, 2002.
- DSMC, Systems Engineering Fundamentals, Defense Systems Management College, Fort Belvoir, VA, 1999.
- S.D. Eppinger and V. Salminen, Patterns of product development interactions, Proc 1st ICSED Conf, Glasgow, August 21–23, 2001, pp. 283–290.
- S.D. Eppinger, D.E. Whitney, R.P. Smith, and D.A. Gebala, A model-based method for organizing tasks in product development, Res Eng Des 6(1) (1994), 1–13.
- J.W. Forrester, Industrial dynamics, Productivity Press, Cambridge, MA, 1961.
- J.W. Forrester, World dynamics. Productivity Press, Cambridge, MA, 1973.
- M. Frank, Cognitive and personality characteristics of successful systems engineers, Proc 10th Annu INCOSE Conf, Minneapolis, MN, 2000, pp. 765–773.
- L. Hatcher, A step-by-step approach to using the SAS® system for factor analysis and structural equation modeling, SAS Institute, Cary, NC, 1994.
- IEEE, Standard for Application and Management of the Systems Engineering Process, IEEE Standard 1220, New York, 1994.
- INCOSE Systems Engineering Handbook, INCOSE-TP-2003-016-02, Version 2a, Seattle, WA, 2004.
- A. Kossiakoff and W.N. Sweet, System engineering management, Wiley, Hoboken, NJ, 2003.
- R. Likert, A technique for the measurement of attitudes, Arch Psychol 140 (1932), 1–55.
- J.M. Lyneis and D.N. Ford, System dynamics applied to project management: A survey, assessment, and directions for future research, Syst Dyn Rev 23(2–3) (2007), 157–189 [Special Issue: Exploring the Next Great Frontier: System Dynamics at 50, Guest Editor: John D. Sterman].
- NASA, NASA systems engineering handbook, NASA-SP-6105, Houston, TX, 1995.
- J. Nunnally, Psychometric theory, McGraw-Hill, New York, 1978.
- Office of Government Commerce, Managing successful projects with PRINCE2, Washington, DC, 2005.
- A.P. Sage and W. Rouse, Handbook of systems engineering and management, Wiley, Hoboken, NJ, 2009.
- A. Sharon, D. Dori, and O. de Weck, Is there a complete project plan? A model-based project planning approach, Proc 18th Annu INCOSE Conf, Singapore, 2009.
- A. Sharon, V. Perelman, and D. Dori, A project-product lifecycle management approach for improved systems engineering practices, Proc 18th Annu INCOSE Conf, Utrecht, The Netherlands, 2008.
- D.V. Steward, Systems analysis and management: Structure, strategy, and design, Petrocelli Books, New York, 1981.
- N.P. Suh, The principles of design, Oxford University Press, Oxford, 1990.



Amira Sharon holds a B.Sc. in Mechanical Engineering (1992), an M.Sc. in Industrial Design (1997), and a Ph.D. in Information Management Engineering (2010), all from Technion, Israel Institute of Technology. Her professional career spans the areas of mechanical engineering, programming, systems engineering, data management of large scale complex systems development, leading systems engineering methodologies development, and management of large-scale projects. Her research interests include conceptual modeling of complex systems, systems engineering management, and combining systems engineering with project management in large-scale complex project-product systems. Amira Sharon is a member of the International Council on Systems Engineering (INCOSE) and a member of the executive committee of the INCOSE_IL, the Israeli Chapter of INCOSE.



Olivier L. de Weck is Associate Professor of Aeronautics and Astronautics and Engineering. He holds an industrial engineering degree from ETH Zurich (1993) and Ph.D. in Aerospace Systems Engineering from MIT (2001). Before joining MIT he was a liaison engineer and later engineering program manager on the F/A-18 aircraft program at McDonnell Douglas (1993–1997). His research interests, teaching emphasis, and professional experience are mainly in two areas: Systems Engineering for Changeability and Commonality and Space Exploration Logistics. Professor de Weck is an Associate Fellow of AIAA, the winner of the 2006 Frank E. Perkins award for excellence in graduate advising, and the recipient of the 2007 AIAA MDO TC outstanding service award. He won two best paper awards at the 2004 INCOSE Systems Engineering Conference, held the Robert Noyce Career Development Professorship from 2002 to 2005, and co-advised the best MIT System Design and Management thesis in 2005. He has over 100 journal and conference publications in the area of systems engineering and space systems design for exploration and communications. His research has been funded by GM, NASA, BP, JPL, Arvin Meritor, DARPA/AFRL and the Alfred P. Sloan Foundation. Professor de Weck is a member of the International Council on Systems Engineering (INCOSE) and the American Society of Engineering Education (ASEE). He served as the General Chair for the 2nd AIAA MDO Specialist Conference in May 2006. He serves as a faculty mentor to a number of student teams.



Dov Dori is Information and Systems Engineering Professor and Head of the Enterprise System Modeling Laboratory at the Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, and Research Affiliate at the Engineering Systems Division, Massachusetts Institute of Technology, where he lectures on a regular basis. In 1999–2001 and 2008–2009 he was Visiting Professor with the Engineering Systems Division, the Department of Aeronautics and Astronautics, and Sloan Business School at MIT. His research interests include conceptual modeling of complex systems, systems architecture and design, and software and systems engineering. Professor Dori has invented and developed Object-Process Methodology (OPM), presented in his 2002 book. He won the Technion Klein Research Award for OPM, the Hershel Rich Innovation Award for OPCAT, the OPM supporting software, and the Dudi Ben Aharon Research Award for Document Image Understanding. Professor Dori has founded OPCAT Ltd., which develops an OPM modeling support environment. He initiated and headed the series of international conferences on Model-Based Systems Engineering held at Technion, Israel in 2007 and 2009, and at George Mason University, VA, in 2010. Professor Dori has authored over 200 publications. He is Fellow of the International Association for Pattern Recognition and Senior Member of IEEE and ACM and member of INCOSE.