Managing Complex Project Process Models with a Process Architecture Framework

Tyson R. Browning
Neeley School of Business
Texas Christian University
TCU Box 298530
Fort Worth, TX 76129 USA
t.browning@tcu.edu

May 23, 2013

Forthcoming in the
International Journal of Project Management

Acknowledgement: This paper is based on an earlier one presented at the First International Conference on Modelling and Managing Engineering Processes (Browning, 2010a). The author is grateful for helpful comments from several anonymous reviewers.

Managing Complex Project Process Models with a Process Architecture Framework

Abstract

Especially in large, complex projects, various aspects of process (activity network) information reside in separate models and diagrams that can become unsynchronized over time. Prior research has introduced the concept of a process architecture framework (PAF), which provides a solution by tying all the models and diagrams together in a single, rich process model with many views, where each view presents a subset of model information. This paper advances that work by (1) proposing an expandable PAF structure that organizes 27+ new and existing views, (2) suggesting examples of three new views that align well with specific concerns of users, and (3) presenting insights to guide the development of new views. Thus, this paper takes further steps towards the development of a PAF that provides at once both simplicity and completeness for project managers and other users of process models and project management information systems.

Key Words: Process modeling; process views; process architecture framework; project management information systems; complexity management; decision support

1. Introduction

Especially in large, complex projects (such as those for the design and development of complex products or services), various aspects of process (activity network) information often reside in separate models and diagrams such as Gantt charts, network diagrams, resource assignment matrices, risk management plans, compliance databases, lessons learned databases, and role and responsibility lists, to name but a few. Because they are built and maintained by different individuals and teams, these models can become unsynchronized over the course of a project. Although some projects have sophisticated information systems to manage many aspects of their process information, even these systems do not yet meet the needs of many users inside and outside the project. Because of these shortcomings, such users many construct their own diagrams, models, spreadsheets, reporting templates, and tracking systems for process information. However, the present danger is that the various models—which contain a great deal of overlapping information—will fall out of synchronization because of their development by disparate organizational units with different information, assumptions, and concerns. For example, a project manager might use a software tool to plan and schedule work, but the list of activities in that tool may fall out of sync with a list of risk management activities kept by designated risk manager, or a list of evidences of process compliance may become disconnected from a standard process kept by process auditors or assessors.

One way to address these issues is to consolidate all of the information about the work done in a project (a project's process information) into a single, rich model with varied *views* (Browning, 2009). A view extracts and displays a subset of a model's attributes and assumptions with an arrangement of symbols, tables, graphs, or other diagrams or depictions (Browning and Ramasesh, 2007). Figure 1 shows an example of three common views drawn from the information in a single, more complex model of a process. Whereas a complex process model might contain some information that only a few types of users care about—such as how an activity is performed on other projects, when its documentation was last updated, who "owns" it, etc.—a particular user will usually only need a small subset of those data to address a specific concern. A scheduler, for example, might care only about the activity's predecessors, duration, and alternative modes (such as opportunities for crashing) and might therefore prefer the Gantt chart and network diagram views. A project manager might care more about the status of activities and deliverables, using views two and three in Figure 1. By extracting different views from the process model, it is possible to provide each of these users with a customized filter that shows them the subset of the model they need. From their own perspective, each is working with his or her own model, but, behind the scenes, all are in fact using a single, common model and therefore "drawing from the same well." Thus, a view offers a particular lens or portal through which to build and/or access certain aspects of a rich process model—a model whose immensity would otherwise cause "information overload" for most users. The use of views from a single model enables the integration and synchronization of the vast amount of information useful for describing, documenting, and managing project work. Management tools such as Microsoft Project® provide a basic form of this capability by allowing a user to toggle back and forth among Gantt chart, network diagram, and tabular views of project activities and resources. In such tools, the data reside in a database, not in any of the particular views, which are reconstructed each time they are accessed by pulling the latest information from the database. However, users do not interact directly with the database. Instead, they input and output data through one or more of the available views. Some users prefer to work in the tabular data entry mode (although each of the available tables only accesses a subset of the database's elements) while others prefer to click on visual elements and pull up dialog windows for data entry. This multi-view approach can be generalized and extended to a much greater portion of project management information, starting with the process (activity network), through the use of a *process architecture framework* (PAF), an organized collection of views of a complex process model.

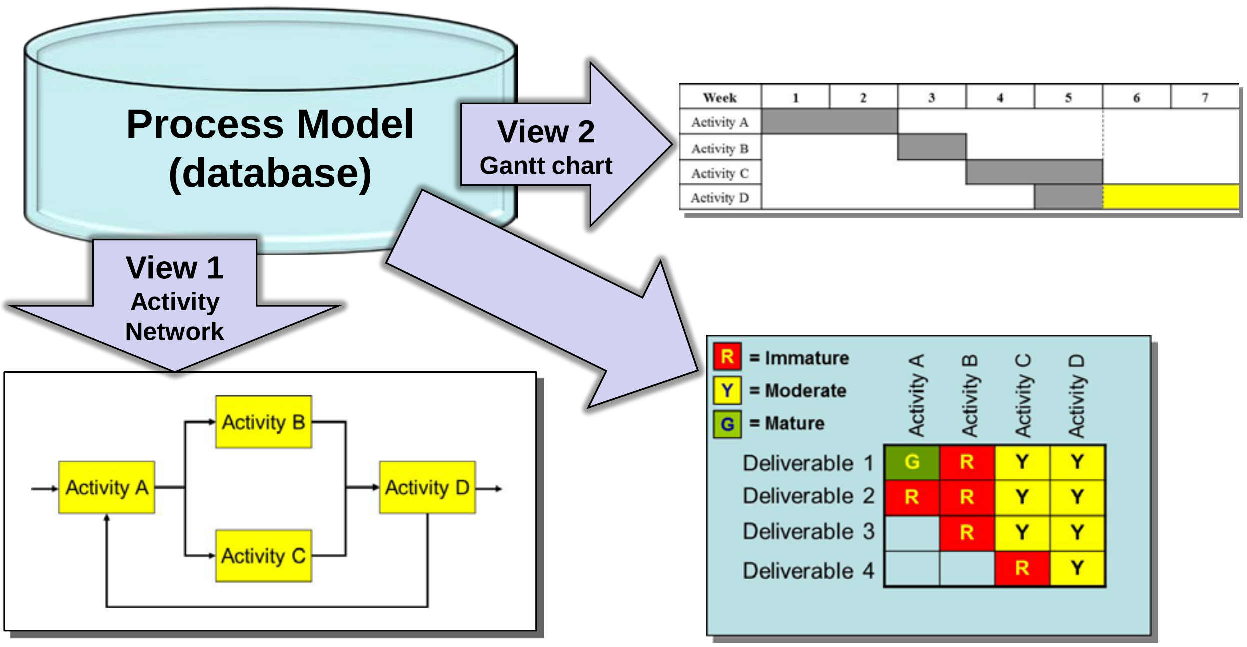

Figure 1: An example of three views extracted from a single, richer model of a process

A PAF is helpful both for assimilating and disseminating the information in a complex process model. Using a synchronous portfolio of PAF views has the potential to greatly enhance project managers' capabilities and prevent unwanted surprises due to data disconnects. But which views should a PAF provide? That depends on who uses process information ("stakeholders") and what concerns they have. Each view in a PAF might seek to align with a particular concern (or category of concerns) by displaying all of the process model data relevant to that concern while excluding irrelevant information. However, a recent study (Browning, 2010b) of both literature and practice found substantial misalignment between 28 concerns and 15 views in terms of 56 information attributes in process models (Table 1).[1] Most problematically, a variety of key concerns from five types of users[2] were not well supported by *any* of the 15 common views examined! Hence, additional and better views of process model information would seem to be needed. Improved views can be a significant driver of innovation in system design (Alexander, 1964; Keller et al., 2005; Keller et al., 2006b; Schätz et al., 2002; Simon, 1996; Zachman, 1987), product development decisions (Krishnan and Ulrich, 2001), and decision support systems in general (Basu et al., 1997), so it is logical to infer that they could be beneficial to project managers and stakeholders as well.

Table 1: Process model concerns, views, and information attributes (adapted from (Browning, 2009), which explains each of these items, and (Browning, 2010b))

Concerns (Purposes)	Views	Process/Activity (Work Package) Attributes	Deliverable (Work Product) Attributes
• Define standard and preferred activities • Define standard deliverables and quality standards • Define standard handoffs and structure standard work flows • Define standard tools and templates • Define standard staffing, roles, responsibilities, and skills • Visualize, understand, analyze, and improve processes • Identify "ripple effects" of process changes • Organize knowledge about work • Tailor the standard process to suit project requirements • Filter activities and deliverables (by hardware vs. software, project size and phase, contract type, etc.) • Associate processes with elements of the project's work breakdown structure (WBS) • Identify appropriate activities and deliverables for the project • Import deployed process activities into a project scheduling tool • Define deployed deliverables and quality levels • Choose tools and templates • Set project schedule and secure formal commitments • Identify skill (or clearance) gaps in the workforce • Estimate project time, cost, quality, and risks • Allocate resources • Visualize planned work flows and integration points • Assign activity roles and responsibilities (staffing) • Monitor project status in terms of activities and deliverables • Renegotiate commitments where necessary • Access knowledge about activities, tools, and deliverables • Deposit lessons learned • View practices relevant to a given standard • Confirm performance of requisite practices • Confirm production of appropriate deliverables	• Process flowchart –Network diagram –PERT chart –Activity-on-node diagram • Gantt Chart • Design Structure Matrix (DSM) • Graphical Evaluation and Review Technique (GERT) Diagram • Textual Narrative • IDEF0 Diagram • IDEF3 Diagram • State Diagram –Event graph –Markov chain –Data flow diagram –Directed graph • Create-Read-Update-Delete (CRUD) Table • Value Stream Map • Supplier-Input-Process-Output-Customer (SIPOC) Diagram –IPO diagram • Entry-Task-Validation-Exit (ETVX) Diagram • Extended Event-driven Process Chain (eEPC) Diagram • Responsibility Assignment Matrix (RAM) –RACI chart • Work Product Standard (WPS) database record	• Name • Parent • Constituents ("Children") • Mode • Shadowing • Deployment • Version Number • Brief Description • Inputs • Outputs • Entry Criteria • Exit Criteria • Verifications • Standard Process Metrics • Deployed Process Metrics • Tools • Standard Roles • Deployed Roles • Basis for Requirement • Rules • References • Standard Risks • Deployed Risks • Narrative Description • Tailoring Guidance • System Identification Number • WBS Element Association • Master Owner • Standard Owner • Deployed Owner • Change History • Change Notifications	• Name • Parent • Constituents ("Children") • Mode • Shadowing • Deployment • Version Number • Brief Description • Suppliers • Customers • Key Criteria and Measures of Effectiveness • Requirements • Acceptance Criteria • Standard Process Metrics • Deployed Process Metrics • Format • Medium • Artifact • Rules • References • Narrative Description • Tailoring Guidance • System Identification Number • WBS Element Association • Change History • Change Notifications

[1] Although the term "purpose" was used in prior literature on PAFs, this paper uses the term "concern" (synonymous in this context) for greater consistency with ISO 42010 (ISO, 2011). Note that ISO 42010 uses the term "stakeholder" for any user with a concern and distinguishes a "viewpoint" as containing the specifications for a "view." This paper does not emphasize the term "stakeholder" because of its wider meaning in project management (i.e., project stakeholders), nor does it delve into distinctions between viewpoints and views. Nevertheless, the terminology and approach in this paper are not inconsistent with ISO 42010.

[2] These users (stakeholders) were project managers and team leaders; owners of (multi-project) standard processes; project planners and schedulers; engineers, designers, and other team members; and process auditors, assessors, and appraisers.

Since earlier work (Browning, 2009) provides theoretical motivation and grounding for PAFs, as well as extensive literature review, this paper merely contributes a step forward in PAF development by proposing an organizational structure for the views, demonstrating three new views tailored to particular concerns, and offering guidelines for view development.

2. Background

A complex project involves a myriad of activities that depend on each other in varied ways to accomplish the project's overall result. Many have studied complexity in project management (e.g., Baccarini, 1996; Bosch-Rekveldt et al., 2011; Geraldi et al., 2011; Tatikonda and Rosenthal, 2000; Vidal et al., 2011; Whitty and Maylor, 2009; Xia and Lee, 2004). Many researchers devise their own definitions of complexity, yet some widely acknowledged characteristics of a system's complexity include its interconnectedness (Simon, 1962), its number of components and relationships (Kauffman and Levin, 1987), and the amount of information required to describe its number of distinct possible states (Bar-Yam, 1997). In the project scheduling literature, several definitions of network complexity depend on the number of activities and dependencies (especially non-redundant arcs). This paper does not seek or depend upon a particular definition or measure of complexity; it merely asserts that the processes (activity networks) used to plan and manage projects can be complex because they represent an enormous amount of information about what work to do, who will do it, and when, where, and how to do it.

What seems complicated to one observer may not seem so to another. Tang and Salminen (2001) distinguished complexity, an objective characteristic of a system, from complicatedness, a subjective, observer-dependent phenomenon (sometimes called cognitive complexity). In their example, "relative to a manual transmission, a car's automatic transmission has more parts and more intricate linkages. It is more complex. To drivers, it is unquestionably less complicated, but to mechanics who must fix it, it is more complicated." The internal complexity of a system can be more or less understood by and hidden from observers, who will in turn deem a fixed amount of complexity as less or more complicated. To an extent observers can reduce complicatedness (while complexity remains constant) by learning about a system and by regulating the presentation of system information in terms of type and amount.

Complex system structure and behavior cannot be fully understood from a single perspective. For example, a house is built using a combination of blueprints, elevations, bills of materials, schedules, cost accounts, and other diagrams (Zachman, 1987). Each group participating in the project (architects, surveyors, contractors, framers, plumbers, electricians, carpenters, masons, roofers, etc.) requires and prefers certain diagrams (or portions thereof) to direct their work. Contractors keep a schedule, a bill of materials, and a list of subcontractors. Electricians use wiring diagrams with particular symbols representing outlets, switches, etc., while the pipe routing diagrams used by plumbers have a different set of symbols and labeling conventions. Nevertheless, all of the work must be done in a harmonious way, and if any of the diagrams do not agree it will lead to problems. In larger and more complex and novel projects than building a house, even more perspectives are involved, the amount of information is much greater, and the possibility and consequence (risk) of disconnects increases. No one manager can "do it all in their head" or ensure that all of the project's participants have the same understandings or bases for decisions. This often leads to a dilemma for managers: on one hand, a large and often overwhelming amount of information must be shared across the project, and this information must be kept up to date and in agreement, while on the other hand each participant in the project needs only a fraction of this information and would prefer to have it provided in a relatively simple, customized format.

An *architecture framework* (AF) is a tool for integrating and managing information about a system in a way that addresses both complexity and complicatedness. Many models can be overly simplistic and fail to account for the complexities of a system. An AF seeks to model a complex system in much greater breadth and detail, seeking a more complete description of its elements, relationships, and behaviors. Yet, this makes the system model larger, more complex, and, in its entirety, more complicated. On the other hand, an AF presents its vast store of information through a collection of relatively simple and uncomplicated views, through which users of the model can input, analyze, or output deliberately chosen subsets of the information. A PAF organizes and synchronizes the views, much like getting the blind men to combine their experiences of the elephant in the old Asian fable (Linton, 1878). AFs therefore provide a solution to the classic tension in managerial models between completeness and simplicity (Little, 1970).

AFs were originally developed for information systems (Zachman, 1987) and have since been applied to other products such as defense systems (DoD, 2009; MoD, 2005), software development (Kruchten, 1995), enterprise information systems (e.g., Tang et al., 2004; TOG, 2006), enterprise architectures (Iyer and Gottlieb, 2004), and space systems (Richards et al., 2007). The ISO 42010 (ISO, 2011) standard now governs basic requirements for AFs, and several companies now provide software tools for building, storing, rendering, and analyzing AF information and views. The emergence and popularity of AFs for a variety of complex products attests to their utility.

However, work on AFs to date has focused on the architectures of complex *products*, such as software, hardware, and so-called "systems of systems." Recent work (Browning, 2009) made the case for analogous applications of AFs to the *processes* of complex projects such as engineering design and product development. These processes involve and integrate many disciplines and perspectives, thus increasing the challenge of the associated managerial and integration tasks. For example, a project manager may have to rely on several process views to support managerial concerns and decisions. She might use a design structure matrix (DSM) (Eppinger and Browning, 2012) to sequence tasks and a Gantt chart (Gantt, 1919) to schedule their start and finish times and track the percentage of each task completed. Meanwhile, she might use earned value management (e.g., Fleming and Koppelman, 2000) or risk value method (Browning et al., 2002) charts to plan and track overall project progress and expenses and technical performance measures. She might also use a responsibility assignment matrix (PMI, 2013) to link people to tasks and/or deliverables and other domain mapping matrices (DMMs) (Danilovic and Browning, 2007) to map requirements to people, activities, and tools. Each of these diagrams contains some unique and common information. The project manager may rely on several individuals to develop and update each of these views, but without careful coordination and verification it would be easy for inconsistencies to emerge. Thus, because a project process is a complex system (Browning et al., 2006), it could benefit from a *process* AF (PAF) that organizes and integrates the superset of information spanning all of the views pertaining to it.

A PAF also benefits other users of process information besides project managers. Models of project processes may be built with any of several potential concerns in mind—e.g., project visualization; project planning, execution, and control; or the development of project and organizational capabilities, including continuous improvement, knowledge management, training, or compliance (Browning and Ramasesh, 2007). However, a model built for one of these concerns may not be useful or properly fit for another (Browning et al., 2006). One size does not fit all—a concept whose violation has led to a great many misunderstandings and failed implementations in practice. For example, a model built to help achieve an ISO certification may not be helpful for day-to-day project management, and a model built for detailed task scheduling may not be useful for training new employees. On the other hand, in places where the concept of tailored models for specific contexts has been embraced, a variety of models emerges, which increases the chances of the redundancy and synchronization problems previously mentioned. By capturing the broadest set of information about the project's process, as useful even to those with a multi-project perspective, a PAF increases the richness of a project process model while providing a variety of views tailored to the various users.

Note that a PAF directly addresses only one crucial domain in a project, the *process* subsystem. Other critical domains (or subsystems) include the project's result (e.g., a *product* or service design), the *organization* of people and teams that will perform the activities (in the process domain) to get the result, the *tools* that the people and teams will use to perform the activities, and the overarching requirements and *goals* the project should achieve (Browning et al., 2006; Danilovic and Browning, 2007). Just as current AFs (e.g., DoD, 2007; MoD, 2005; TOG, 2006) focus on the *product* domain, and the PAF developed herein focuses on the *process* domain, it would also be useful to consider the possibilities of AFs for the *organization*, *tool*, and *goal* domains. Indeed, some work along these lines has already begun in the organization domain (Williams and Stracener, 2012). Eventually, it might be beneficial to combine all five of these domains to arrive at a full *project* AF, which would provide a holistic plan for a full project management information system. However, this paper focuses only on the *process* domain.

3. An Initial Proposal for Structuring a PAF

Version 1.5 of the U.S. Department of Defense Architecture Framework (DoDAF) (DoD, 2009) consists of 29 views in four categories:

- Overall views: two general views that provide overview and summary information (e.g., scope, purpose, users, and intended uses) as well as a dictionary of terminology (i.e., an ontology);
- Operational views: nine views that address various aspects of how the system (project result) will be used, representing functional, organizational, and information flows;
- Systems and Services views: 16 views that describe the product (project result) itself, such as components, interfaces, data flow, performance parameters, and much more; and
- Technical Standards views: two views that note applicable (and potentially applicable) standards.

While the DoDAF is oriented towards *product* (project result) rather than process systems, it has received some of the most extensive development (through several major versions) of any AF to date, and it is logical to infer that a PAF could benefit from a similar structural scheme, such as the following proposal:

- *Overall* views: general views that provide overview and summary information (e.g., scope, purpose, users, and intended uses) as well as a dictionary of terminology (i.e., an ontology) of the project's process;
- *Operational* views: views that address various aspects of project planning and process execution, such as guidelines for process tailoring and deployment, scheduling, commitment gathering, monitoring and control, etc.;
- *Content* views: views that describe the process itself, especially activities, deliverables (work products), the activity network, resource requirements, and performance metrics; and
- *Technical Standards* views: views that note applicable (and potentially applicable) standards, benchmarks, and/or best practices (e.g., ISO 9000, CMMI[3], a company's internal standards, etc.).

Table 2 maps 28 concerns (purposes) of project process models (first column of Table 1) against 56 information attributes (last two columns of Table 1). Each cell contains an integer from zero (blank) to two indicating whether the attribute of information is essential, merely helpful, or unnecessary to addressing a particular concern. The rows and columns of Table 2 have been reordered based on a clustering analysis to find groups of related concerns and attributes. The first shaded cluster in the upper-left consists of seven concerns that use most of the activity attributes. The second shaded cluster in the upper-right includes nine concerns that use most of the deliverable attributes. These two clusters overlap. The third shaded cluster in the middle-left associates nine concerns with moderate to small needs for activity attributes and almost no needs for deliverable attributes. The last set of concerns, grouped at the bottom, uses a moderate number of attributes from both objects. This analysis suggests a further breakdown of *Content* views into two categories, rich (generalized) and targeted (specialized). Rich views would contain many information attributes to service the more expert users, while targeted views would support concerns that require a much smaller subset of attributes. Table 2 also suggests a further subdivision of the rich views into those focused either on activities or deliverables.

[3] Capability Maturity Model—Integrated is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Table 2: Concern / attribute clusters (adapted from (Browning, 2010b))

Essential:	2
Helpful:	1
Unneeded:	0

Concern (Purpose)	Process/Activity Object Attributes																														
	Name	Mode	Deployment	Parent	Constituents ("Children")	Inputs	Outputs	Deployed Owner	Exit Criteria	Deployed Process Metrics	Deployed Roles	Brief Description	Entry Criteria	Tools	Standard Owner	Deployed Risks	Verifications	Rules	Shadowing	Standard Process Metrics	Basis for Requirement	Master Owner	Change Notifications	Standard Risks	WBS Element Association	Standard Roles	Narrative Description	Change History	References	Tailoring Guidance	Version Number
Define standard and preferred activities	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	1	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
Organize knowledge about work	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
Access knowledge about activities, tools, and deliverables	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2	1	1	2	2	1	1
Identify "ripple effects" of process changes	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	1	2	1
Deposit lessons learned	2	2	2	2	2	2	2	2	2	2	2	1	2	1	2	2	2	2	2	1	2	1	2	2	1	1	2	2	2	2	
Tailor the standard process to suit project requirements	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	1	2	1	2	2	2	1	1		2	1
Visualize, understand, analyze, and improve processes	2	2	2	2	2	2	2	1	2	2	1	1	2	1	1	2	2	2	2	2	2	1	2	2	1	1	1	2	1	1	
Define standard deliverables and quality standards																															
Define deployed deliverables and quality levels																															
Confirm production of appropriate deliverables																															
Confirm performance of requisite practices	2	2	2	2	2	2	2	2	2	1	1	1	1		1		2	2			2						2	2	1		2
Estimate project time, cost, quality, and risks	2	2	2	2	2	2	2			2		1				2				2				2							
Allocate resources	2	2	2	2	2	2	2			2		1		2						1											
Import deployed process activities into a project scheduling tool	2	2	2	2	2	2	2	2		2	2																				
View practices relevant to a given standard	2	2	2	2	2			2							2						2	2									
Identify skill (or training) gaps in the workforce	2	2	2	1	1			1			2	1		1													1				
Assign activity roles and responsibilities (staffing)	2	2	2					2			2	1		1												2					
Define standard staffing, roles, responsibilities, and skills	2	2									1	1			1							1				2					
Define standard tools and templates	2	2										1		2	1							1					1				
Identify appropriate activities and deliverables for the project	2	2	1	2	2	2	2		1	2	2	1	1						2	1					2						
Set project schedule and secure formal commitments	2	2	2	2	2	2	2	2	2			1	2										2								
Renegotiate commitments where necessary	2	2	2	2	2	2	2	2	2	2		1	2			2							2								
Monitor project status in terms of activities and deliverables	2	2	2	2	2	2	2	2	2	2		1				2															
Define standard handoffs and structure standard work flows	2	2	1	1	1	2	2		2				2		2				1			2									
Choose tools and templates	2	2	2					1	2		2	1	1	2			2	2									1		2	1	
Filter activities and deliverables	2	2	2	2	2							1							1						2						
Visualize planned work flows and integration points	2	2	2	2	2	2	2	2																							
Associate processes with WBS elements	2	2		2	2														1						2						
Column count:	50	50	41	40	40	34	34	31	27	26	24	23	23	20	20	19	18	18	17	17	17	17	17	16	16	15	15	15	13	13	9
Column rank:	1	1	3	4	4	6	6	8	9	10	11	12	12	14	14	16	17	17	19	19	19	19	19	24	24	26	26	26	29	29	31

Concern (Purpose)	Deliverable Object Attributes																								
	Name	Mode	Parent	Constituents ("Children")	Suppliers	Consumers	Deployment	Key Criteria and Measures of Effectiveness	Requirements	Acceptance Criteria	Deployed Process Metrics	Format	Medium	WBS Element Association	Rules	Shadowing	Artifact	Brief Description	Change Notifications	Tailoring Guidance	Change History	Standard Process Metrics	Narrative Description	References	Version Number
Define standard and preferred activities																									
Organize knowledge about work	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2
Access knowledge about activities, tools, and deliverables	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	1	2	1	1	2	2
Identify "ripple effects" of process changes	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	2	2	2	2	1	1	1
Deposit lessons learned	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	1	2	2	2	1	2	2	
Tailor the standard process to suit project requirements	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	2	1	2	1		1
Visualize, understand, analyze, and improve processes	2	2	2	2	2	2	2	2	2	2	2	1	1	1	2	2	1	1	2	1	2	2	1	1	
Define standard deliverables and quality standards	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Define deployed deliverables and quality levels	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	1		2		1	1		
Confirm production of appropriate deliverables	2	2			2	2	2	2	2	2	1				2			1			2		2		2
Confirm performance of requisite practices																									
Estimate project time, cost, quality, and risks																									
Allocate resources																									
Import deployed process activities into a project scheduling tool																									
View practices relevant to a given standard																									
Identify skill (or training) gaps in the workforce																									
Assign activity roles and responsibilities (staffing)																									
Define standard staffing, roles, responsibilities, and skills														2											
Define standard tools and templates																									
Identify appropriate activities and deliverables for the project	2	2	2	2	2	2	2	1	1	1	2	1	1	2			1	1				1			
Set project schedule and secure formal commitments	2	2	2	2	2	2	2	2	2	2	2	2	2		1		1	1	2						
Renegotiate commitments where necessary	2	2	2	2	2	2	2	2	2	2	2								2						
Monitor project status in terms of activities and deliverables	2	2	2	2	2	2	2	2	2	2	2						1								
Define standard handoffs and structure standard work flows	2	2	2	2	2	2		2	2	2		2	2			2	2								
Choose tools and templates	2				1	1		1	1	1		1	1		1		1			1			1	1	
Filter activities and deliverables	2	2	2	2			2							2		1		1							
Visualize planned work flows and integration points	2	2	1	1	2	2												1							
Associate processes with WBS elements	2	2	2	2			2							2		2		1							
Column count:	36	34	31	31	31	31	30	28	28	28	24	21	21	21	20	18	18	17	17	15	15	14	14	11	10
Column rank:	1	2	3	3	3	3	7	8	8	8	11	12	12	12	15	16	16	18	18	20	20	22	22	24	25

Concern (Purpose)	Row Count (Activity Object)	Row Count (Deliverable Object)	Percent Populated (Activity Object)	Percent Populated (Deliverable Object)	Total Row Count
Define standard and preferred activities	57	0	92%	0%	57
Organize knowledge about work	61	49	98%	98%	110
Access knowledge about activities, tools, and deliverables	56	46	90%	92%	102
Identify "ripple effects" of process changes	58	45	94%	90%	103
Deposit lessons learned	54	44	87%	88%	98
Tailor the standard process to suit project requirements	53	41	85%	82%	94
Visualize, understand, analyze, and improve processes	49	40	79%	80%	89
Define standard deliverables and quality standards	0	49	0%	98%	49
Define deployed deliverables and quality levels	0	38	0%	76%	38
Confirm production of appropriate deliverables	0	26	0%	52%	26
Confirm performance of requisite practices	36	0	58%	0%	36
Estimate project time, cost, quality, and risks	23	0	37%	0%	23
Allocate resources	20	0	32%	0%	20
Import deployed process activities into a project scheduling tool	20	0	32%	0%	20
View practices relevant to a given standard	18	0	29%	0%	18
Identify skill (or training) gaps in the workforce	14	0	23%	0%	14
Assign activity roles and responsibilities (staffing)	14	0	23%	0%	14
Define standard staffing, roles, responsibilities, and skills	10	2	16%	4%	12
Define standard tools and templates	10	0	16%	0%	10
Identify appropriate activities and deliverables for the project	25	26	40%	52%	51
Set project schedule and secure formal commitments	23	31	37%	62%	54
Renegotiate commitments where necessary	27	24	44%	48%	51
Monitor project status in terms of activities and deliverables	23	23	37%	46%	46
Define standard handoffs and structure standard work flows	20	26	32%	52%	46
Choose tools and templates	23	14	37%	28%	37
Filter activities and deliverables	14	14	23%	28%	28
Visualize planned work flows and integration points	16	11	26%	22%	27
Associate processes with WBS elements	11	15	18%	30%	26

These insights motivated the PAF structure proposed in Table 3. Of the two dominant categories of views, the *Operational* views will appeal especially to project planners and managers, while the *Content*

Table 3: An initial proposal for the PAF structure

View Type	View	General Description
Overall	Overview and Summary Information	Scope, purpose, and intended users and applications of the process (e.g., product development projects, derivative product development, new product development, or detailed design phase)
	Integrated Dictionary	Terminology definitions (ontology)
Operational	Activity Schedule	Information about the scheduled timing of each activity in the process; a variety of views are possible, but the basic ones are: (a) Gantt chart and (b) table of activity start times, durations, and stop times
	Process Capability	Views of process performance expectations (e.g., duration, cost, quality, value at risk, etc.) and tradeoff options (e.g., time-cost tradeoff curves); would include views used to simulate "what if" explorations
	Project State	Summary views of project status versus goals in terms of cost (vs. budget), schedule (vs. deadline), and key aspects of technical performance (vs. requirements), including prevailing levels of uncertainty and risk
	Responsibility Assignment Matrix	Links process activities to organizational units (which could be further modeled in the organization domain)
	Resource Allocation Map	Shows how many of each of various types of resources are required to execute each activity (see Figure 4 for an example)
	Resource Loading Profiles	Shows the overall need for each type of resource versus the amount available
	Commitment Network	Identifies where agreements and commitments have been formalized with those responsible for particular activities and deliverables and signals where delays will prompt renegotiated commitments
	Process Constraints	Identifies business rules that constrain the process in general and in a particular project
	Contingencies	Views depicting predetermined responses to anticipated scenarios and their implications
	Monitoring	Identifies key metrics to be collected and used in managing process execution
	Earned Value	Depiction of metrics such as planned value, earned value, cost-schedule indices, etc.
	Risk Value	Depiction of metrics such as value at risk due to cost, schedule, and various attributes of technical performance; technical performance measure tracking charts
	Tailoring Guidance	Lays out the possibilities for modifying the standard process description as it is applied to a particular project and aids project planners in selecting the appropriate subset of activities
	(additional views as needed)	The PAF will grow and develop as new views are proposed, developed, and found to be useful; users might also be able to customize various views by toggling or rearranging content
Content	Activity Inspector*	Rich views of activities (work packages) and their attributes; a variety of views are possible, but the basic ones are: (a) activity attribute inspector form and (b) SIPOC diagram; a textual narrative could also be used (see Figure 3 for an example)
	Deliverable Inspector*	Rich views of deliverables (work products) and their attributes; a variety of views are possible, but the basic one is the deliverable attribute inspector form (see Figure 2 for an example)
	Work Breakdown Structure	Hierarchical breakdown of process into sub-processes and activities; could be provided as an indented list, a traditional organization chart, and/or a directory tree
	Deliverable Breakdown Structure	Hierarchical decomposition of deliverables could be provided as an indented list, a traditional organization chart, and/or a directory tree
	Activity Network	Depict the process flows resulting from the interlinking of activities and their input/output deliverables; a variety of views are possible, but the basic ones are: (a) flowchart and (b) design structure matrix
	Process Improvement	Planned or desired changes to the process to improve its overall value
	Technology Roadmap	Emerging technologies that may affect future development of the process
	Tools	Links process activities to tools
	Standard Roles	Links process activities to roles, skills, and necessary or desired attributes of organizational units
	Technical Performance Measures	Standard attributes of technical performance and their measures
	(additional views as needed)	The PAF will grow and develop as new views are proposed, developed, and found to be useful; users might also be able to customize various views by toggling or rearranging content
Technical Standards	Technical Standards Profile	Listing of standards that apply to activities and deliverables
	Technical Standards Forecast	Description of emerging standards and potential impact on current activities and deliverables, within a set of time frames

*Rich views

views will appeal strongly to process owners and improvers (who may have a multi-project perspective). Many more specific views could surely be added to this table, which is not intended to comprehensive. Indeed, it is expected that others will expand and enhance the collection of views in the PAF, and that any particular project would select and use only the subset of views that provides benefits exceeding costs.

4. Three Proposals for New Views

As examples, this section proposes three new views that could augment the traditional views incorporated into the PAF. The first two are rich views in the *Content* category, and the third is a targeted, *Operational* view.

The "goodness" of each view partly depends on its alignment with one or more of the concerns from Tables 1 and 2. This fit is evaluated via the alignment metrics developed in (Browning, 2010b), referred to herein as *concern-view alignment* (CVA). CVA is a scalar index normalized as a percentage, where 0% indicates complete misalignment (the inclusion of none of the attributes needed to address the concern and all of the unneeded ones) and 100% implies perfect alignment (the inclusion of all needed attributes and no unneeded ones)—so higher scores are better. Since CVA depends on the chosen number and type of attributes used in the analysis, it is mainly useful in a relative sense for comparing views. Table 4 compares CVA results from 15 views (Browning, 2010b) with the newly proposed views (in the right-most columns), highlighting the best fitting (highest scoring) view for the concern in each row. While there are several 0% scores, indicating that a view is maximally unfit for a concern, a salient finding in (Browning, 2010b) was the lack of any really good scores. (The best CVA, 74%, was obtained by the Extended Event Process Chain diagram (provided by the ARIS software tool—Scheer, 1999) when used for the concern "Visualize planned workflows and integration points.")

The first proposal for a new, rich view of deliverable attributes is shown in Figure 2, preceded by some comments on its features. While a direct view into a database of attributes (e.g., the existing "WPS database record" view) also provides access to this information, an interactive form such as the one shown in Figure 2 provides this capability more elegantly. One of the right-most columns in Table 4 shows results for the new "Deliverable Inspector Form," which exhibits the best CVA for two concerns and could be modified through further customization to provide complete alignment (CVA = 100%) with concerns 2-10 in Table 2. (Note that the concerns are listed in different orders in Tables 2 and 4.) In addition to providing the appropriate attributes, the Deliverable Inspector Form demonstrates the further value of a meaningful arrangement of its content (something that CVA does not measure). Different users may find alternative arrangements more intuitive, and the capability to customize the view by user could be provided to an extent.

Along similar lines, Figure 3 exhibits a proposal for a rich view of activity attributes, which in this particular format provides the highest CVA measures for three concerns (see Table 4). These scores could be further improved by customizing the views to particular concerns, especially concerns 1-7 in Table 2. It is also interesting to notice that, when both the activity and deliverable inspector forms are used together, they provide the best CVA for half (14) of the concerns. This corroborates earlier work that proposed the development of user interfaces similar to those shown in Figures 2 and 3 (Negele et al., 1999).

Table 4: CVA metrics of new and traditional views (adapted from (Browning, 2010b))

	Concern-View Alignment (CVA) of Information Attributes	Views																			
		WPS database record	Extended EPC diagram	RAM	Textual narrative	IDEF0 diagram	IDEF3 diagram	GERT diagram	Process flowchart	SIPOC diagram	Value Stream Map	ETVX diagram	Gantt chart	DSM	CRUD Table	State diagram	Activity Inspector Form	Deliverable Inspector Form	(Both forms)	Resource Allocation Map	Average
Concerns	Organize knowledge about work	65	60	55	67	61	58	54	57	60	55	57	56	58	52	52	68	66	**84**	59	60
	Identify "ripple effects" of process changes	59	61	56	66	62	59	54	58	60	56	57	57	58	53	52	66	63	**82**	60	60
	Visualize, understand, analyze, and improve processes	58	60	47	67	60	57	49	55	59	57	55	54	56	53	52	67	60	**82**	58	58
	Define standard and preferred activities	0	39	55	71	63	66	58	64	59	48	63	62	61	35	54	**84**	0	60	61	53
	Define standard deliverables and quality standards	83	30	0	13	9	0	0	0	10	10	0	0	4	20	0	0	**85**	59	0	17
	Define standard handoffs and structure standard work flows	46	60	37	42	49	46	53	46	55	37	49	42	44	56	54	38	51	**64**	39	48
	Define standard tools and templates	0	22	46	51	45	39	32	41	52	18	33	34	40	28	35	**56**	0	51	52	36
	Define standard staffing, roles, responsibilities, and skills	0	40	**73**	51	35	39	32	41	44	18	33	34	40	28	35	56	0	51	38	36
	Tailor the standard process to suit project requirements	57	62	56	65	63	60	55	59	61	56	58	57	59	53	52	66	63	**83**	61	60
	Set project schedule and secure formal commitments	57	61	31	43	47	44	52	41	53	41	43	40	39	55	54	39	58	**68**	51	48
	Identify appropriate activities and deliverables for the project	48	67	36	51	54	53	52	49	54	59	30	46	43	55	54	48	54	**72**	59	52
	Define deployed deliverables and quality levels	76	32	0	15	10	0	0	0	11	11	0	0	4	21	0	0	**79**	57	0	17
	Choose tools and templates	40	31	33	52	34	21	21	18	34	11	37	15	21	42	55	42	39	**61**	35	34
	Filter activities and deliverables	27	49	30	38	46	42	37	35	51	24	10	27	29	44	29	37	36	**58**	44	36
	Associate processes with WBS elements	28	50	30	33	43	38	37	35	43	24	11	27	29	44	29	32	37	**54**	35	35
	Estimate project time, cost, quality, and risks	0	44	32	51	58	64	62	58	55	63	28	57	61	39	29	64	0	53	**84**	48
	Import deployed process activities into a project scheduling tool	0	55	53	48	63	69	64	62	59	57	35	61	54	41	30	62	0	52	**80**	50
	Allocate resources	0	48	33	52	70	69	64	62	63	63	29	61	54	41	30	66	0	55	**100**	51
	Identify skill (or training) gaps in the workforce	0	38	**62**	54	45	44	46	40	54	31	21	36	39	25	32	58	0	51	61	39
	Renegotiate commitments where necessary	45	62	31	46	51	49	52	45	54	49	43	46	47	55	54	46	43	**63**	59	50
	Monitor project status in terms of activities and deliverables	46	64	32	46	52	50	53	46	55	50	35	47	48	56	41	44	45	**64**	61	49
	Visualize planned work flows and integration points	31	**74**	36	41	62	56	59	50	63	42	31	49	48	60	29	44	33	60	61	49
	Assign activity roles and responsibilities (staffing)	0	35	**62**	44	33	32	27	27	42	15	21	21	26	25	32	48	0	42	49	31
	Confirm performance of requisite practices	0	40	44	65	59	63	63	59	57	44	61	60	52	37	56	**74**	0	58	62	50
	Confirm production of appropriate deliverables	**71**	25	0	18	12	0	0	0	13	6	0	0	5	22	0	0	62	48	0	15
	View practices relevant to a given standard	0	30	56	49	53	**61**	42	53	52	28	31	51	46	24	31	53	0	45	53	40
	Access knowledge about activities, tools, and deliverables	63	59	51	67	62	59	54	58	60	56	57	57	58	53	52	68	66	**85**	60	60
	Deposit lessons learned	61	60	51	67	59	56	49	55	58	54	54	53	55	53	52	65	61	**81**	57	58
	Average:	34	49	40	49	49	46	44	43	50	39	35	41	42	42	38	50	36	**62**	51	
	Maximum:	83	74	73	71	70	69	64	64	63	63	63	62	61	60	56	84	85	85	100	

As a third example, Figure 4 shows a targeted, Operational view, the "Resource Allocation Map." Again, targeted views support concerns that require fewer information attributes. They may also adopt some aspects of traditional views. This view targets the concern "Allocate Resources," with which it has a perfect CVA of 100% (Table 4): it provides exactly the information attributes needed to support resource allocation decisions without contributing to cognitive overload by including superfluous information.

It is important to reiterate that CVA does not tell the whole story. Just because a view includes and excludes exactly the right attributes to address a concern does not necessarily mean that it does a good job of presenting those attributes to a user—i.e., of arranging the attributes in a way that supports quick situation assessment and decision making. Style matters as well as substance. Although CVA does not quantify this aspect of the fit between concern and view, it is noted as an important area for future research.

Deliverable (Work Product) Inspector Form

- Provides a deep look at the attributes of a deliverable (work product), which can be selected from the hierarchical list on the left (only one level of the hierarchy is shown, but a directory tree structure could be used to show multiple levels)
- When selected, the name of the deliverable appears in the "Name" text box, and many of its other attributes are populated in other text boxes
- The main, center portion of the form is arranged similarly to an input-process-output (IPO) or supplier-IPO-customer (SIPOC) diagram, with buttons for adding and deleting entries
- The lower portion of the form provides further details for any activity or sub-deliverable clicked once in the lists; clicking twice on an activity or deliverable opens a new form specific to that activity or deliverable
- A user can edit some of all of these fields depending on his or her permissions
- This particular example of the form can be altered to include or exclude attributes

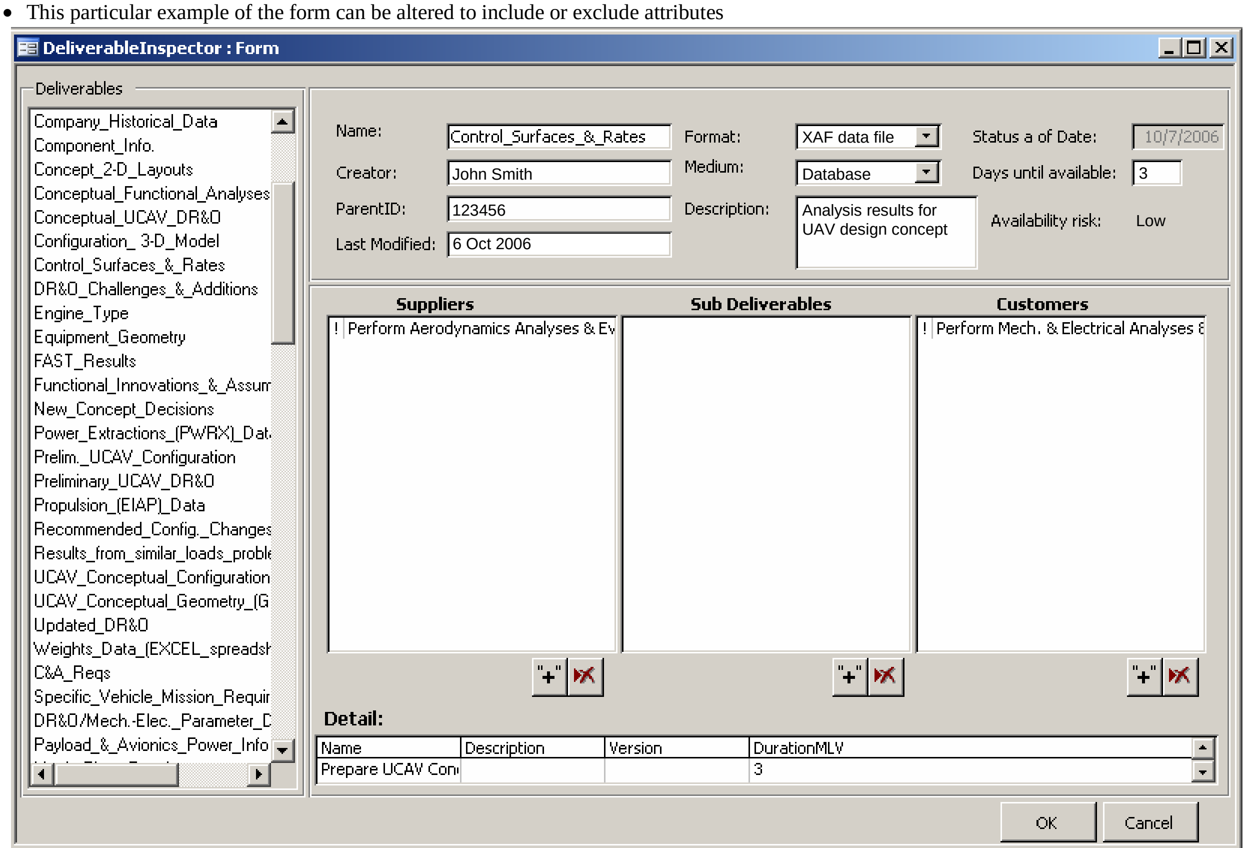

Figure 2: A rich view of deliverable (work product) attributes

5. Some Guidelines for Developing Views

Preliminary research suggests the following nine general guidelines for the development of new views:

1. Maximize concern-view alignment (CVA). A practical view must suit a concern, meaning that it should contain just the right information to support a particular decision or purpose, without any superfluous or extraneous content. It is essential to present the right information to support a concern without causing distraction or "information overload" (Browning, 2010b).

Activity Inspector Form

- Provides a deep look at the attributes of an activity, which can be selected from the hierarchical list on the left
- When selected, the name of the activity appears in the "Name" text box, and its other attributes are populated in the other text boxes
- The main, center portion of the form is arranged similarly to an input-process-output (IPO) diagram, with buttons for adding and deleting entries
- The lower portion of the form provides further details for any activity or deliverable clicked once in the lists; clicking twice on an activity or deliverable opens a new form specific to that activity or deliverable
- A user can edit some of all of these fields depending on his or her permissions
- This particular example of the form can be altered to include or exclude attributes

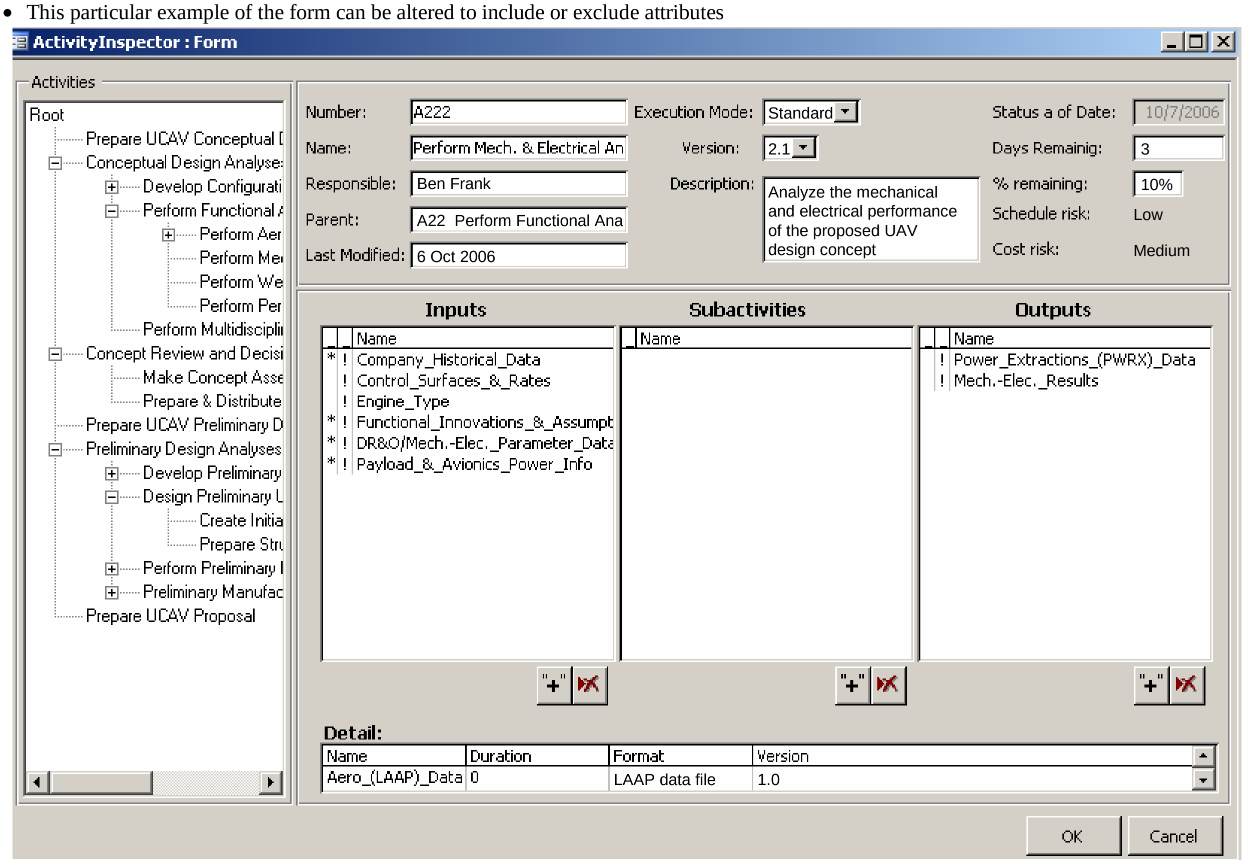

Figure 3: A rich view of activity attributes

2. Content arrangement also matters in a view. Having just the right information (as measured by CVA) is a start, but that information must also be arranged in an appropriate and intuitive way. The idea is to get the key information needed for a concern to come across clearly in a view. Formatting is a user-dependent aspect, however, as what seems intuitive to one user might seem complicated to another. Further research is needed to explore the efficacy and popularity of various view formats. Although several studies have touched on aspects of information visualization in the context of project processes (e.g., Keller et al., 2006a; Killen and Kjaer, 2012; Maurer et al., 2009), further work should organize such results with respect to the particular concerns listed in Table 1.
3. The size of a typical computer monitor should be considered. In the interviews with project managers and process model users described in (Browning, 2010b), some commented on the difficulty of comprehending a large process flowchart (that would cover the wall of a conference room if printed)

when looking at it "through a straw" (their computer monitor). While the capability to zoom in and out is helpful, a view should ideally be legible on a computer screen without requiring the user to scroll around or zoom in and out too much. This is facilitated by process modelers decomposing their process models into no more than 10 or so activities at each hierarchical level. Hierarchies of activities and deliverables (captured by the parent-child attributes) can help manage the complexity of a large number of activities and relationships.

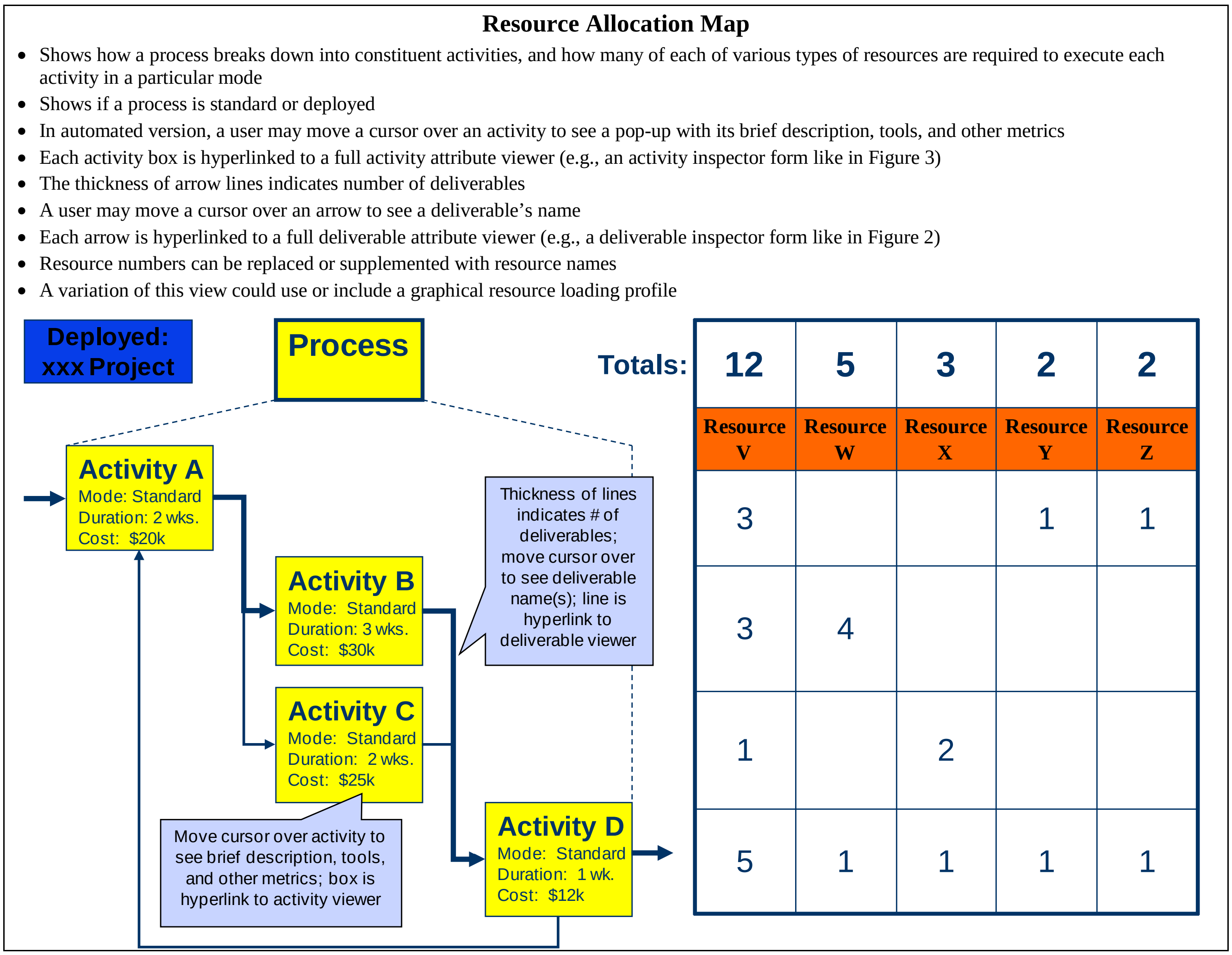

Figure 4: A targeted view for the concern "Allocate resources"

4. Views should be linked to further information from the overall model (and other views). A mouse cursor, pop-up boxes, and links can be used as mechanisms for accessing additional information without crowding a view. For example, the activity boxes in Figure 4 are linked to additional information about the activities. Rolling the cursor over a box brings an automatic pop-up with information on the activity attributes needed for the concern, while actually clicking on the box brings a full activity inspector window like the one in Figure 3. However, as the presence of links and pop-

ups may not be immediately obvious to a user, it would be problematic to obscure the essential information this way.

5. The synchronization of multiple views requires the distillation of a common set of attributes (such as the set in Table 1) that reside in a common database (or a linked set of databases) and which the views draw upon as a common standard. This database is the actual process model, while the views are lenses through which certain aspects of the model are seen, evaluated, and used for various concerns (Browning, 2009).
6. A view may provide two-way access to a process model. That is, a view can be used for both input and output of process information, for building as well as using a process model. The form in Figure 3, for example, not only shows attributes but may also allow for their editing. Specific views might even be created to guide concerns involving the initial building of process models (e.g., a process modeling "wizard"). Using a view as an input device would probably require the implementation of access privileges and perhaps some capability to verify and validate inputs automatically. Certain users could have permissions to access and modify particular attributes. This would address concerns about data security and the sharing of process information with partner companies and suppliers.
7. It is preferable to generate views automatically, on demand, rather than statically. For example, instead of storing a view as a portable document format (PDF) file that must be updated manually any time the process information changes, a view should be generated dynamically using the latest information at the time of access. This helps keep all users synchronized and all managerial decisions based on common information. Yet, it calls for views that are easy to generate automatically. Complex flowcharts, for example, may not be easy to generate in an easily useable way, because the algorithms for laying out flow paths in a visually appealing way (e.g., without crossing flow lines) can be challenging to implement. However, a view with a more standard layout, such as a design structure matrix (DSM), requires less sophisticated algorithms to automate its display.
8. Given the need for rapid updating and dissemination of a process model and its views, an intranet-based (web-based) application tool has advantages. Internet browsers provide a common access point available to all users and seem to provide a useful paradigm for implementing views (Arita et al., 2007; Sabbaghian et al., 1998).
9. The customization of views, even down to the individual user level, is possible and often beneficial. Two different users may prefer to arrange the same attribute information in different formats. Thus, to an extent, users should be able to rearrange content to suit their understanding. Parsimony of views is not essential or even desirable. A software application for viewing process model information would do well to facilitate the customization of views. Over time, particularly useful views would emerge and gain grassroots popularity. Researchers could study which views emerge as especially popular or

beneficial in practice.

6. Conclusion

This paper contributes to the ongoing development of a PAF, a rich model of a project's activity network and its attributes as seen through a collection of views. The specific contributions are the organizational structure for a PAF proposed in Table 3, the three new views (Figures 2-4) developed to address specific concerns, and the guidelines for view development offered in §5. Certainly, many more steps are needed before a useful PAF is available, but hopefully this work will prompt further discussions and spark new efforts towards that end—which will itself be just a beginning, since a PAF is intended to evolve as new concerns, views, and attributes become important. In particular, future research is needed on the following aspects and issues: content versus form (or substance versus style) in views, additional views for critical concerns, the automated synchronization of views, and the development of supporting software tools. Further definition of the PAF's specifications, rules, and conditions is needed in conformance with ISO 42010.

Meanwhile, practitioners can benefit from the way of thinking about and managing complexity that AFs provide. Although the software tools are not yet available to integrate all of their project management models and views, managers can emphasize the integration and synchronization of project information to help move in that direction. As mentioned earlier, software tools such as Microsoft Project® already provide some of the basic capabilities of a PAF by storing process information in a background database and allowing users to input and output data through alternative views (customizable tables, customizable Gantt chart, network diagram, resource loading profile, etc.). This approach can be expanded to include the concerns of other users of process data (e.g., as listed in the first column of Table 1) and still other views (e.g., as listed in the second column of Table 1, as proposed in §4, and as will be developed in the future). Even as they await this expansion of capability and functionality on the part of vendors of project management information systems, project managers can seek to identify the broad set of users (stakeholders) of process data, emphasize its coordination and synchronization across the organization, and consider which subsets of process data users want to see (and how they want to see it) to support their decisions.

7. References

Alexander, C., 1964. Notes on the Synthesis of Form. Harvard University Press, Cambridge, MA.

Arita, Y., Nakayama, N., Awata, Y., 2007. Development Process Visualization and Project Management. Fujitsu Scientific & Technical Journal, 43, 1, 97-104.

Baccarini, D., 1996. The Concept of Project Complexity - A Review. Int. J. of Project Management, 14, 4, 201-204.

Bar-Yam, Y., 1997. Dynamics of Complex Systems. Addison-Wesley, Reading, MA.

Basu, A., Blanning, R.W., Shtub, A., 1997. Metagraphs in Hierarchical Modeling. Management Sci., 43, 5, 623-639.

Bosch-Rekveldt, M., Jongkind, Y., Mooi, H., Bakker, H., Verbraeck, A., 2011. Grasping Project Complexity in Large Engineering Projects: The TOE (Technical, Organizational and Environmental) Framework. International Journal of Project Management, 29, 6, 728-739.

Browning, T.R., 2009. The Many Views of a Process: Towards a Process Architecture Framework for Product Development Processes. Systems Engineering, 12, 1, 69-90.
Browning, T.R., 2010a. Further Development of a Process Architecture Framework, in: *Proc. of the 1st Int. Conf. on the Modelling and Management of Engineering Processes (MMEP)*, Cambridge, U.K., July 19-20, pp. 29-40.
Browning, T.R., 2010b. On the Alignment of the Purposes and Views of Process Models in Project Management. Journal of Operations Management, 28, 4, 316-332.
Browning, T.R., Deyst, J.J., Eppinger, S.D., Whitney, D.E., 2002. Adding Value in Product Development by Creating Information and Reducing Risk. IEEE Transactions on Engineering Management, 49, 4, 443-458.
Browning, T.R., Fricke, E., Negele, H., 2006. Key Concepts in Modeling Product Development Processes. Systems Engineering, 9, 2, 104-128.
Browning, T.R., Ramasesh, R.V., 2007. A Survey of Activity Network-Based Process Models for Managing Product Development Projects. Production and Operations Management, 16, 2, 217-240.
Danilovic, M., Browning, T.R., 2007. Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices. International Journal of Project Management, 25, 3, 300-314.
DoD, 2007. DoD Architecture Framework, Version 1.5, U.S. Dept. of Defense, DoD Architecture Framework Working Group, Apr. 23.
DoD, 2009. DoD Architecture Framework, Version 2.0, U.S. Dept. of Defense, DoD Architecture Framework Working Group.
Eppinger, S.D., Browning, T.R., 2012. Design Structure Matrix Methods and Applications. MIT Press, Cambridge, MA.
Fleming, Q.W., Koppelman, J.M., 2000. Earned Value Project Management, 2nd ed. Project Management Institute, Upper Darby, PA.
Gantt, H.L., 1919. Organizing for Work. Harcourt, Brace and Howe, New York.
Geraldi, J., Maylor, H., Williams, T., 2011. Now, Let's Make It Really Complex (Complicated): A Systematic Review of the Complexities of Projects. Int. Journal of Operations & Production Management, 31, 9, 966-990.
ISO, 2011. Systems and Software Engineering -- Architecture Description. International Organization for Standardization (ISO), Geneva, Switzerland.
Iyer, B., Gottlieb, R.M., 2004. The Four-Domain Architecture: An Approach to Support Enterprise Architecture Design. IBM Systems Journal, 43, 3, 587-597.
Kauffman, S.A., Levin, S., 1987. Towards a General Theory of Adaptive Walks on Rugged Landscapes. Journal of Theoretical Biology, 128, 1, 11-45.
Keller, R., Eckert, C.M., Clarkson, P.J., 2005. Viewpoints and Views in Engineering Change Management, in: *Proc. of the 2nd Workshop on Complexity in Design and Engineering*, Glasgow, Scotland, Mar 10-12, pp. 188-192.
Keller, R., Eckert, C.M., Clarkson, P.J., 2006a. Matrices or Node-link Diagrams: Which Visual Representation is Better for Visualising Connectivity. Information Visualization, 5, 62-76.
Keller, R., Flanagan, T.L., Eckert, C.M., Clarkson, P.J., 2006b. Two Sides of the Story: Visualising Products and Processes in Engineering Design, in: *Proceedings of the 10th International Conference on Information Visualisation*, London, England, Jul. 5-7, pp. 362-367.
Killen, C.P., Kjaer, C., 2012. Understanding Project Interdependencies: The Role of Visual Representation, Culture and Process. International Journal of Project Management, 30, 5, 554-566.
Krishnan, V., Ulrich, K.T., 2001. Product Development Decisions: A Review of the Literature. Management Science, 47, 1, 1-21.
Kruchten, P., 1995. Architectural Blueprints—The “4+1” View Model of Software Architecture. IEEE Software, 12, 6, 42-50.
Linton, W.J., 1878. Poetry of America: Selections from One Hundred American Poets from 1776 to 1876. William Clowes and Sons, London.
Little, J.D.C., 1970. Models and Managers: The Concept of a Decision Calculus. Management Science, 16, 8, B466-B485.
Maurer, M., Braun, T., Lindemann, U., 2009. Information Visualization for the Structural Complexity Management, in: *Proceedings of the 19th Annual Symposium of the International Council on Systems Engineering (INCOSE)*, Singapore, July 20-23
MoD, 2005. Ministry of Defence Architectural Framework Overview, Version 1.0, UK Ministry of Defence, Document MODAF-M09-002, Aug. 31.
Negele, H., Fricke, E., Schrepfer, L., Härtlein, N., 1999. Modeling of Integrated Product Development Processes, in: *Proceedings of the 9th Annual International Symposium of INCOSE*, Brighton, UK, June 6-11, pp. 1253-1261.

PMI, 2013. A Guide to the Project Management Body of Knowledge, 5th ed. Project Management Institute, Newtown Square, PA.
Richards, M.G., Shah, N.B., Hastings, D.E., Rhodes, D.H., 2007. Managing Complexity with the Department of Defense Architecture Framework: Development of a Dynamic System Architecture Model, Massachusetts Institute of Technology, Engineering Systems Division, Working Paper ESD-WP-2007-09, Feb.
Sabbaghian, N., Eppinger, S., Murman, E., 1998. Product Development Process Capture & Display Using Web-Based Technologies, in: *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, San Diego, CA, Oct. 11-14, pp. 2664-2669.
Schätz, B., Pretschner, A., Huber, F., Philipps, J., 2002. Model-based Development, Institut für Informatik, Technische Universität München, Technical Report TUM-I0204
Scheer, A.-W., 1999. ARIS - Business Process Modeling, 2nd ed. Springer, New York.
Simon, H.A., 1962. The Architecture of Complexity. Proc. of the American Philosophical Society, 106, 6, 467-482.
Simon, H.A., 1996. The Sciences of the Artificial, 3rd ed. MIT Press, Cambridge, MA.
Tang, A., Han, J., Chen, P., 2004. A Comparative Analysis of Architecture Frameworks, School of Information Technology, Centre for Component Software and Enterprise Systems, Swinburne University of Technology, Technical Report SUTIT-TR2004.01, Aug. 25.
Tang, V., Salminen, V., 2001. Towards a Theory of Complicatedness: Framework for Complex Systems Analysis and Design, in: *Proc. of the 13th International Conf. on Engineering Design (ICED)*, Glasgow, Scotland, Aug.
Tatikonda, M.V., Rosenthal, S.R., 2000. Technology Novelty, Project Complexity, and Product Development Project Execution Success: A Deeper Look at Task Uncertainty in Product Innovation. IEEE Transactions on Engineering Management, 47, 1, 74-87.
TOG, 2006. The Open Group Architecture Framework (TOGAF) 8.1.1. The Open Group.
Vidal, L.-A., Marle, F., Bocquet, J.-C., 2011. Measuring Project Complexity using the Analytic Hierarchy Process. International Journal of Project Management, 29, 6, 718-727.
Whitty, S.J., Maylor, H., 2009. And then came Complex Project Management (revised). International Journal of Project Management, 27, 3, 304-310.
Williams, J.L., Stracener, J.T., 2012. First Steps in the Development of a Program Organizational Architectural Framework (POAF). Systems Engineering, forthcoming.
Xia, W., Lee, G., 2004. Grasping the Complexity of IS Development Projects. Communications of the ACM, 47, 5, 69-74.
Zachman, J.A., 1987. A Framework for Information Systems Architecture. IBM Systems Journal, 26, 3, 276-292.