Accuracy of Aggregate Data in Distributed Project Settings: Model, Analysis and Implications

NITIN R. JOGLEKAR, Boston University
EDWARD G. ANDERSON, University of Texas at Austin
G. SHANKARANARAYANAN, Babson College

We examine the management of data accuracy in inter-organizational data exchanges using the context of distributed software projects. Organizations typically manage projects by outsourcing portions of the project to partners. Managing a portfolio of such projects requires sharing data regarding the status of work-in-progress residing with the partners and estimates of these projects' completion times. Portfolio managers use these data to assign projects to be outsourced to partners. These data are rarely accurate. Unless these data are filtered, inaccuracies can lead to myopic and expensive sourcing decisions. We develop a model that uses project-status data to identify an optimal assignment of projects to be outsourced. This model permits corruption of project-status data. We use this model to compute the costs of using perfect versus inaccurate project-status data and show that the costs of deviation from optimal are sizable when the inaccuracy in the data is significant. We further propose a filter to correct inaccurate project-status data and generate an estimate of true progress. With this filter, depending on the relative magnitudes of errors, we show that accuracy of project-status data can be improved and the associated economic benefit is significant. We illustrate the improvement in accuracy and associated economic benefit by instantiating the model and the filter. We further elaborate on how the model parameters may be estimated and used in practice.

Categories and Subject Descriptors: I.6. [**Simulation and Modeling**]; I.6.5. [**Model Development: Modeling Methodologies**]; H.1. [**Information Systems**]; H.1. [**Models and Principles**]; H.1.m. [**Miscellaneous**]

General Terms: Design, Measurement, Management

Additional Key Words and Phrases: Aggregate project planning, data quality, Kalman filtering, value of data

ACM Reference Format:
N. R. Joglekar, E. G. Anderson, and G. Shankaranarayanan. 2013. Accuracy of aggregate data in distributed project settings: Model, analysis and implications. ACM J. Data Inf. Qual. 4, 3, Article 13 (May 2013), 22 pages.
DOI: http://dx.doi.org/10.1145/2458517.2458521

1. INTRODUCTION

Data quality is associated with business value [Even and Shankaranarayanan 2007]. That is, improving data quality and associated reliability can reduce business risks and opportunity costs. Better data allows for improved decision making and consequently leads to improved efficiency and performance [Banker and Kaufman 2004]. In recent years, sharing analytical data has become a foundational business practice in many industries, enabled by the Internet and wireless technologies. Many, if not most, projects

Authors' addresses: N. R. Joglekar, Operations Management Department, Boston University School of Management, Boston, MA; E. G. Anderson, McCombs School of Business, University of Texas at Austin, Austin, TX; and G. Shankaranarayanan (corresponding author), TOIM Division, Babson College, Babson Park, MA; email: gshankar@babson.edu.

DOI: http://dx.doi.org/10.1145/2458517.2458521

are executed in distributed settings, relying extensively on sharing data. However, research that examines data quality in settings where data are exchanged across departmental and organizational settings suggested in Shankaranaryanan et al. [2003] is in an emergent phase. The early contributions posit high-level frameworks for managing quality [Cai and Shankaranarayanan 2007]. At present, this literature neither addresses the economics associated with poor-quality data nor suggests solutions for improving data quality in inter-organizational settings.

In this article, we examine not only the economic impact of data shared across organizational boundaries but also suggest a solution, using a filter, for improving the quality of the shared data. Our solution uses collaborative filtering that has been used in information systems to improve decision quality in such instances as Yahoo!Movies recommendations [Sahoo et al. 2011]. Specifically, we use the Kalman-Bucy filter [Stengel 1994]. A commonly used filter is exponential smoothing of data [Trigg and Leach 1967; Makridakis and Winkler 1983]. However, the advantage of using Kalman-Bucy filter is that the decisions can be shown optimal under conditions when the data are normally distributed, which is the case in the context we study in this article. Moreover, the application of collaborative filtering, specifically Kalman-Bucy filtering, for improving data quality has not been fully explored in inter-organizational settings. Our solution demonstrates how collaborative filtering may be used to improve data quality, specifically, the accuracy of data. A major issue that has been emphasized for managing data quality in inter-organizational settings is the need for cooperation between the participating organizations. Our solution does not mandate a symbiotic effort between the two (or more) participating organizations (we acknowledge that cooperation will certainly help but is not required). By implementing the filtering solution we compute the improvements in costs associated with the decisions. Our results clearly show that the benefit of filtering the data is significant when the data accuracy is poor (i.e., noise level is high or signal-to-noise ratio is low) and practically insignificant when the condition is reversed.

Data quality may be measured along a number of different quality dimensions such as accuracy, completeness, consistency, timeliness, and reliability [Wang and Strong 1996]. Our focus in this article is on accuracy. Accuracy is defined in literature as the correctness of a data value compared to some baseline value that is known to be correct [Ballou et al. 1998; Pipino et al. 2002; Fisher et al. 2003]. Accuracy is difficult to measure as the baseline value is typically unknown at the time of measurement (e.g., the accuracy of forecasted demand for a product for a future period is difficult to gauge until the actual demand for this period is known, much later). Hence accuracy is typically estimated using statistical analysis of historical data to reveal estimation errors. Approaches to improve accuracy such as data cleansing can be useful if correct data values are known [Mookerjee et al. 1995; Hernandez and Stolfo 1998]. In our research scenario, accurate values are not obtainable. Similarly, data tracking and statistical process control can help estimate the extent of inaccuracy, but do not help with improving accuracy [Redman 1996]. Techniques such as total data quality management [Wang 1998], can improve accuracy at source, but do not mitigate the effects of inaccurate data. Here we explore a situation where correcting the accuracy at source is difficult and expensive, and managers are forced to deal with inaccurate data. We propose a solution to improve data accuracy by smoothing out inaccuracies through filtering. We compare the costs of making decisions with inaccurate data and with corrected data obtained by improving accuracy through filtering.

1.1. Research Context

The context in which we examine data quality in inter-organizational settings is the management of projects in distributed project settings. In today's global environment

projects are typically carried out by multiple teams in distributed settings [Hirscheim et al. 2002]. In such settings, there is typically a primary program office (referred to as the *principal*) and several development partners (called *partners*), who may be geographically distributed. For instance, when the principal organization is faced with a large project, for example, development of a product such as Microsoft Office [Cusumano and Selby 1995] or an SAP implementation [Jones and Price 2004], they often break these large projects into smaller, more manageable subprojects or tasks and distribute them among principal and partners. Other, smaller projects are distributed as a single, stand-alone task for completion in one firm (principal or a partner). The principal will decide to outsource some projects to one or more partners depending on the principal's own costs and capacity as well as the progress data (number of projects the partner has in house and the estimated completion times for the projects) available about each partner. The effective allocation of projects among principal and partners determines the success of the projects and the economic welfare of the principal [Mandel and Engardio 2007; Chen and Bharadwaj 2009].

Thus, the principal is charged with managing the elements of this portfolio (whether stand-alone projects or farmed-out subprojects) in aggregate as a portfolio of projects [PMI-PMBOK 2008]. For the purpose of this article we use the term "project portfolio management" to refer to the aggregate planning and integration. Moreover, the terms "projects" and "tasks" are used interchangeably, to describe the individual elements of this overall portfolio. Effectively distributing projects (or subprojects within a large software project) across multiple partners requires that a majority of the projects are routine, that is, the requirements are well-specified and can be executed either by the principal or by one or more partners [Upton and Staats 2006]. The principal must, however, maintain a certain fraction of the work in house, for example, testing, to keep the customer-related IP and protect itself against optimistic behavior associated with partners [Ford and Sterman 2003]. The principal must also carefully manage the coordination and productive execution of the projects, individually and in aggregation [Gomes and Joglekar 2008].

An assumption behind effectively managing an aggregate project portfolio involving distributed projects (or tasks) is the availability of perfect data about projects (such as progress of work at the partner and estimates of completion time). We use the term "project-status data" to refer to the aggregate data exchanged between the partners and principal for coordination and project execution. Such data do not deal with bugs or dependencies within any one project, but rather with the status of the project as it relates to the overall portfolio. Literature on the management of distributed projects assumes that perfect project-status data about the progress made by the partners is available [PMI-PMBOK 2008]. "Perfect" project-status data are accurate and free of any error due to incorrect estimation of time to complete the work or errors in aggregating data in information systems. This assumption is critical for planning as the principal and partners use up-to-date project-status data to make resource allocation decisions. However, aggregate project-status data are usually far from perfect (accurate) [Caballero et al. 2009]. It is difficult, if not impossible, to correct these errors at source. The completion times of projects (i.e., the time needed to complete projects that are in progress at a partner's site) are created through an estimation process. Errors in estimations can be amplified during aggregation (completion times aggregated across projects to determine completion time for the portfolio). Managers are aware that project-status data are flawed and yet are forced to use them as the basis of their decisions. We examine this problem as a data quality problem in inter-organizational settings. Data quality literature has not examined this issue in depth.

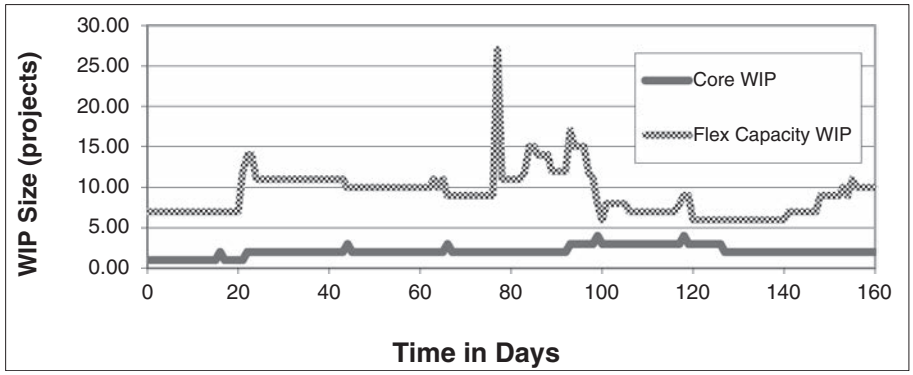

Fig. 1. Amounts of WIP (Work-In-Progress) with core group and partner (flex group).

1.2. Motivating Example

To motivate the problem we examine in this article and to show why aggregate project-status data errors are difficult correct at source, we present an example that illustrates decisions, project-status data, and its quality issues in distributed project planning. This example is based on observations of the aggregate planning and oversight process for the software program office at an insurance firm. In this firm, most product offerings are developed as software features such as graphical interfaces for generating insurance quotations and for updating tools for assessing damages due to natural disasters. Many of these start as "bugs" in existing features. New features are also added based on market needs and customer feedback. Each fix, enhancement, or new addition is bundled into a number of finite-duration projects that can either be executed by one group or assigned to multiple other groups. The principal (software program office) then keeps track of the number of projects that are in progress, which it refers to as "Work-In-Progress" (WIP). Note that when measuring WIP, the principal adjusts each project by its estimated completion status. For example, if they believe a project is 90% complete, then they would count the project as $1 - 0.90 = 0.10$ projects for the purposes of WIP calculation. We also note that the observed WIP is neither smooth nor a monotonic function, because sometimes rework can be revealed in an existing project, and at other times a new project may be added.

A planning team in the principal firm assigns the projects/tasks either to a software development group within the firm (a.k.a. core group) or outsources it to a different software development group that resides in another firm (a partner). This core group has a fixed number of personnel who work on several projects concurrently. The partner has flexible capacity (and is referred to as the *flex* group in Figure 1), as it can further outsource excessive workload. Some of the projects, when finished by the partner, are transferred back to the core group because these projects need to go through integration and/or testing. Figure 1 shows a time series of the WIP, measured in number of projects, for the core group and the partner (flex) over a period of 160 days. The average size of WIP for the principal during the period is 1.7 (standard deviation = 0.61) projects while that of the partner during the period is 7.4 (standard deviation = 2.80) projects.

We interviewed the principal's planning team about their project assignment practices. The project manager to whom a specific project is assigned is primarily concerned with completing this project as quickly as possible (minimizing completion time). Alternately, the planning manager for the overall (aggregate) process is concerned about managing the utilization of capacity and smoothing the workload. These managers pointed out that they always work with a target WIP level. While the targeted WIP

is good for smoothing workload and facilitating multitasking, these managers are also concerned with ensuring that the WIP does not get out of hand. Work done outside the principal firm, particularly with partners (a different outside firm), requires more coordination and has the probability of requiring some integration, and hence for this firm all of the project can never be outsourced in its entirety, as a matter of policy.

Another issue is highlighted by the nature of the data in Figure 1. Since each project's completion time, ex ante, is a random distribution, the actual hours of work ex post required to complete the WIP projects depicted will not be exactly proportional to the series depicted in the table. This fact draws attention to another difficulty in this planning process, highlighted in standard texts on management of development projects (see Meredith and Mantel [2005]): the uncertainty associated with the measurement of WIP and the poor quality project-status data generated. While it is relatively easy to count the number of unfinished projects, the planners cannot fully observe the progress of individual projects, nor can they perfectly estimate any given project's required work hours (i.e., time needed to finish a project in progress). Hence the status of the aggregate WIP is inaccurate. It can, at best, be estimated with uncertainty. However, these inaccurate project-status data are used for allocating new projects that come in for planning.

Such accuracy problems are not unique to a software program office in an insurance firm. Anecdotal evidence from multiple industries also suggests that project-status data are either noisy and/or inaccurate [Dominguez 2006]. For instance, during our fieldwork at the embedded software office of a large electronic product firm, the planning managers raised the issue of accuracy in the project-status data for their projects and posed the question: how much should the firm be willing to spend to clean it? Accordingly, we explore the following questions as central issues in this article: (1) how does the knowledge of status of the progress affect optimal assignments for aggregate planning in distributed work settings? (2) What is the cost impact of imperfect data on the development of aggregate plans in such settings? Finally, (3) can measures be taken to ameliorate the cost impact of imperfect data?

Based on these questions, we make four key contributions. First, we develop a control-theoretic model to understand sourcing decisions for projects in a distributed setting. The model allows us to determine the optimal assignment of outsourced projects using project-status data (the decision). The model also allows us to model corrupt project-status data to simulate errors (such as measurement errors, aggregation errors, etc.). Second, we use the model to explore the effect of cost-structure asymmetry and derive an optimal division of projects between a principal and its partners (cost impact of the decision). This will illustrate that, with perfect project-status data, such a decision will limit projects from overrunning time/resource constraints. It will also show that with random disturbances in demand and reintegration, project plans are relatively stable in their variation with perfect data as compared with using inaccurate data. Third, we determine the value of perfect versus imperfect project-status data under the previously determined "optimal" control rule (what is it worth to fix errors). Fourth, we offer a solution for improving the accuracy of project-status data using filters. We show how the filter allows us to access (timely) real-time project-status data for making outsourcing decisions. We begin the rest of this article with an overview of relevant literature. We then describe the model and derive the optimal control rule. We also describe the filter to smooth out inaccuracies in project-status data. Using numerical data, we instantiate the model and the filter to illustrate their applicability. We analyze the sensitivity of the results to a relevant range of data errors. The errors are characterized using signal-to-noise ratios. The lower the noise, the higher the accuracy and the higher the signal-to-noise ratio. A low signal-to-noise ratio implies higher noise and consequently, lower accuracy. We also examine how filtering the data (and improving

accuracy) may reduce the aggregate planning cost penalties associated with inaccurate data. We further describe how the model parameters may be gathered or estimated to implement the model and the filter in practice. We then discuss the implications of our findings for managing data accuracy of project-status data in distributed project settings. We conclude by presenting the limitations of this research and identifying opportunities for further work.

2. RELEVANT LITERATURE

To lay the foundation for our research, we describe the following areas: value of information (or data) and capacity planning in multiproject settings for sourcing decisions. In addition, our work builds on control-theory literature. We have not elaborated on this area here and refer readers to the section in the appendix titled "Link to Management Science". Our focus is on the economic impact of poor data quality on decisions in distributed project settings, a solution to improve quality, and the economic improvements achieved by improving quality. To the best of our knowledge, very little literature exists addressing the area of data quality in inter-organizational settings. We reviewed research in data quality in the introductory section as we motivated our research problem. We have hence not addressed data quality further. We briefly review the relevant literature on information value. We then treat the two areas of capacity planning and multiproject management as setting the stage for our research. We summarize the preceding literature with the objective of differentiating our research from other similar work.

The literature in information economics differentiates between perfect, imperfect, and incomplete information (data) [Tirole 1993]. Perfect data pertains to a noncooperative setting in which the player whose turn it is to play knows all the actions taken before her turn. With imperfect data, a player does not know what the other player has done so far. Incomplete data describes the situation in which a player does not know the other player's precise characteristics. Our work here differs in an important respect, because it follows the dynamic control literature [Bertsekas 2001] by differentiating between perfect and imperfect data, not on the basis of the structure of a noncooperative game, but rather on whether or not the data are error free. Hence, under this definition of imperfect data, a firm cannot measure even the state of its own internal WIP progress without some uncertainty and without errors.

Literature in software project management has proposed different approaches for estimating software development work: expert judgment, formal estimation models, and a combination of these two. Jorgensen [2007] showed that using expert judgment offers more accurate estimates of development work than formal estimation models as experts use important contextual data in the estimation. The same study also revealed that the combination of the two approaches also offers comparable accuracies. While such techniques can improve the accuracy of the estimates, literature also acknowledges that the best estimates are still not very accurate (e.g., Jorgensen [2007]). We have hence not examined the expert judgment system as a technique to improve quality of project-status data and assume that the project-status data are still not perfectly accurate. As shown later, the applicability of our solution is dependent on how "accurate" are the project-status data. Chiang and Mookerjee [2004] present a management policy in which the appearance of software faults during system construction is used to determine the timing of system integration activities. In this article we address the impact of accurate project data on the optimal management of project resources needed to enable integration.

In summary, the key decision task analyzed is the one to outsource a fraction of projects in distributed project settings. The relevant data, project-status data exchanged between the principal and partners, are inaccurate. It is difficult to fix

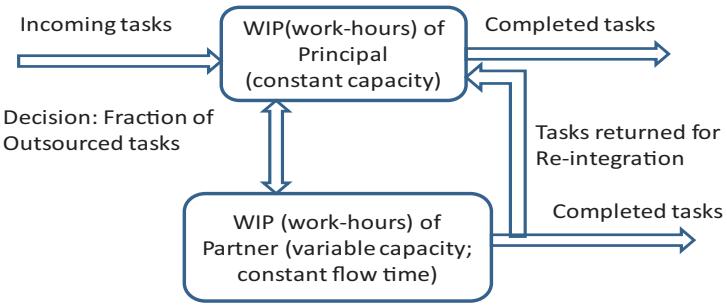

Fig. 2. Base model.

inaccuracies in these data. Next, we develop a model for optimizing sourcing decisions, to study the economic impact of inaccurate data on the decision to outsource. We suggest a filtering solution to reduce inaccuracies.

3. OUTSOURCING DECISIONS: MODEL AND ANALYSIS

Figure 2 models a distributed system as two reservoirs with reflow. The principal has a constant capacity (number of employees) consistent with practice [McGrath 2001]. The demand for new work feeds the principal's WIP. For convenience, but without loss of generality, we now measure the projects, as adjusted by their progress, by the estimated work hours necessary to complete them (i.e., 2 projects of 5000 hours each, each of which is 30% completed, would represent a WIP of 3000 hours).[1] The principal keeps and executes work internally ("in-sourcing") and/or outsources to a partner. The partner has flexible capacity; it can change the number of its employees each period so that it can accept all work given and complete each after a constant flow time, consistent with current practices in software outsourcing. (It is important to note that, without loss of generality, this partner can represent the aggregate characteristics of a number of multiple partners.) Both the principal and the partner can complete work; however, a fraction of the work completed by the partner must pass back to the principal for integration, such as testing, which is performed internally. Each project completed induces a reintegration penalty, which is typical of real-world outsourcing of product and process (including software) development [Parker and Anderson 2003].

3.1. Tracking Variations in WIP

To model this system, let $t \geq 0$ represent time and the index $d \geq 0$ represent the time period. Each period is of duration δ, that is, the time period d begins at time $t = \delta d$ and ends at $t = (\delta+1)d$. Since we are modeling at the aggregate level across a large number of different projects, we model the arrival of each period's numerous projects at the principal as a Poisson process (e.g., Carrascosa et al. [1998])—although this assumption can be relaxed without loss of generality. The project arrival process has an average inter-arrival time of α. The number of hours to complete each project is distributed randomly (with independent, identical distributions) with respect to the work required to complete them. Let i and $\boldsymbol{o}$ be indices that denote in-sourced (i.e., principal) and outsourced (i.e., partner) work. Hence, $\boldsymbol{x_{o,d}}$ and $\boldsymbol{x_{i,d}}$ will represent the number of work hours in progress stocks at the principal and partner, during period $\boldsymbol{d}$, respectively. The size of each WIP

[1]Note that we follow the lead of the embedded software office of the electronics firm mentioned in the Introduction in measuring WIP in hours instead of projects adjusted for their completion. While the two measures are conceptually identical, "hours" simplifies the exposition of discussion of estimation error later in the article.

may or may not be known perfectly due to how well each project's remaining number of hours before completion can be estimated [Meredith and Mantel 2005]. We also assume that the flow of projects outsourced from the principal to the partner is a weighted sum of the two WIPs (which, as we show later, is an optimal policy). We further assume that the completion rate of hours of work at the principal's WIP is constant because it maintains a constant number of employees, $\boldsymbol{c_i}$. Each employee can complete a certain number of hours of work each period. There may be additional variation in the net completion rate due to various factors, such as administrative misallocation, distractions, stages in the project lifecycle, etc. This variation will be represented by a Gaussian random disturbance with constant mean and standard deviation. These variations in employee productivity capture distractions at work, holidays, administrative overhead, etc.

The partner, in contrast, can adjust its number of employees each period. That is, the number of partner employees is proportionate to the number of hours in its WIP $\boldsymbol{x_o}$, that is, $\boldsymbol{c}_{o,d} = \boldsymbol{\kappa\delta x}_{o,d}$. This implies that, on average, each unit of work spends an identical amount of time, that is, $\boldsymbol{\kappa}^{-1}$ hours, in WIP prior to completion, as required. The partner also experiences a Gaussian random disturbance analogous to that at the principal, but with potentially different mean and standard deviation. The reintegration penalty is modeled as a flow of hours into the principal's WIP per period that is proportionate to the completion of work in the partner's WIP. Throughout this article, we will also assume that, over time, the average completion rate of work at each WIP stock is equal to the average arrival rate of new and, for the principal, reintegration work. Without this assumption, the WIPs would either go to infinity or repeatedly approach zero. Both cases have been ruled out in practice by Morrice et al. [2004]. Note, in the firms that inspired our model, planning periods are typically a week in duration, email inboxes fill on a seemingly hourly basis with new projects, and these projects are indeed random. Given these time scales, we assert that these projects accumulate into a WIP stock and that the variation in net flow from this stock of WIP would be governed by the law of large numbers and result in a Gaussian distribution. We present the lemmas (1 through 3), their proofs and theorems in an electronic appendix to formally support this assertion.

3.2. State Equations

Given the assertion in Section 3.1, we pose the outsourcing problem described in the previous section using a continuous time approximation as follows. Let:

$r(t)$ = end-customer demand rate;

c = completion rate at the principal, a constant;

λ = fractional completion rate of WIP per unit time at the partner;

$w_i(t)$ = the net variation in the changes in stock of WIP at the principal, $w_i(t) \sim N(0, \sigma_i^2)$ and $\text{cov}(w_i(t), w_i(t+\varepsilon)) = 0$, for $\varepsilon \neq 0$;

$w_o(t)$ = the net variation in the changes in stock of WIP at the partner, $w_o(t) \sim N(0, \sigma_o^2)$ and $\text{cov}(w_o(t), w_o(t+\varepsilon)) = 0$, for $\varepsilon \neq 0$;

$x_i(t)$ = the work in progress, measured in number of hours of work, at the principal at time *t;*

$x_o(t)$ = the work in progress, measured in number of hours of work, at the partner at time *t;*

ϕ = the fraction of projects completed at the partner requiring reintegration at the principal ($0 \leq \phi < 1$); and

$u(t)$ = the number of projects transferred from the principal to the partner at time *t.* Note when $u(t) < 0$, rein-sourcing of previously outsourced projects occurs.

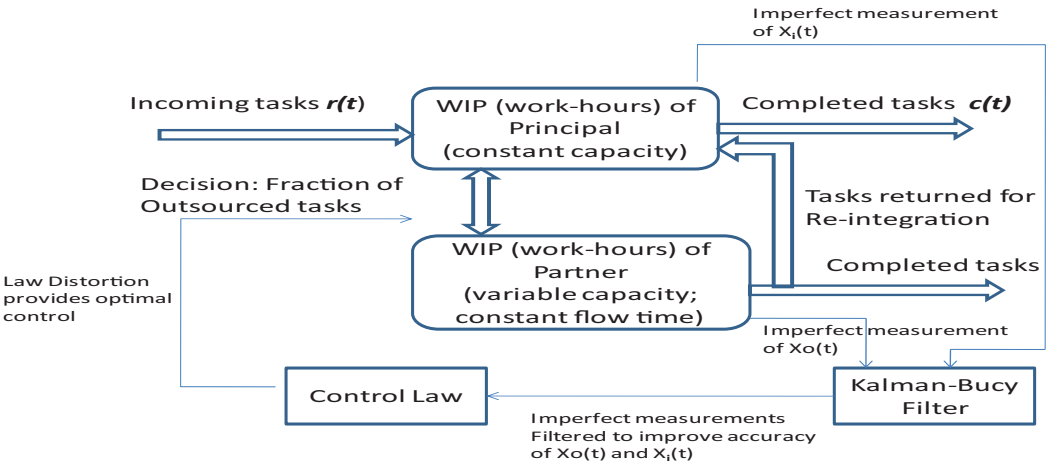

Fig. 3. Closed loop control with inaccurate data.

The system is updated based on the following project balance relationships.

$$\frac{dx_i(t)}{dt} = r(t) - c + w_i(t) - u(t) + \phi[\lambda x_o(t) + w_o(t)] \tag{3.1}$$

$$\frac{dx_o(t)}{dt} = u(t) - [\lambda x_o(t) + w_o(t)] \tag{3.2}$$

Eq. (3.1) tracks the rate of change of the in-sourced work in process. For ease of understanding $\boldsymbol{x_i(t)}$ is the work in progress, measured in number of hours of work, at the principal at time $\boldsymbol{t}$. Referring to the arrows in Figure 3, $\boldsymbol{r(t)}$ are the incoming tasks, $\boldsymbol{c(t)}$ is the completion rate of tasks, $\boldsymbol{u(t)}$ is the fraction of tasks outsourced to partners, and the last term represents the tasks that return to the principal for reintegration. Further, $\boldsymbol{w_i(t)}$ is the net variation because of the rework created at the principal site itself. Analogously, the reader can track Eq. (3.2) which tracks the rate of change of the outsourced work in process. Eqs. (3.1) and (3.2) are linked via the decision variable $\boldsymbol{u(t)}$ and the amount of reintegration work needed determined by the fraction $\boldsymbol{\phi}$.

3.3. Control with Perfect Data

To obtain a reasonable control policy for $\boldsymbol{u(t)}$, we look forward from the present time $t = 0$ seeking to balance WIP, capacity, and capacity adjustment costs. Let:

$\beta_i^2(t)$ = penalty on in-sourced backlog (square is merely for analytic convenience), $\beta_i > 0$;
$\beta_o^2(t)$ = penalty on outsourced backlog (square is merely for analytic convenience), $\beta_o > 0$;
$\tilde{x}_i$ = target in-sourced WIP, $\tilde{x}_i \geq 0$;
$\tilde{x}_o$ = target outsourced WIP, $\tilde{x}_o \geq 0$; and
J = the total penalty function.

As discussed in the illustrative example, the objective for aggregate planning is to smooth out the WIP while accounting for dissimilar cost and integration penalties.

Thus our objective is to

$$\min_{u(t)} J = \lim_{t_f \to \infty} \frac{1}{t_f} \int_0^{t_f} \left\{ u^2(t) + \beta_i^2 \left[X_i(t) - \tilde{X}_i \right]^2 + \beta_o^2 \left[X_o(t) - \tilde{X}_o \right]^2 \right\} dt. \tag{3.3}$$

The first term reflects the quadratic cost associated with the control effort, and the second and third terms account for the quadratic costs associated with deviations from the target in-sourced and outsourced WIP. Fundamentally, the objective function in Eq. (3.3) performs an infinite horizon trade-off between smoothing each stage's work in process (WIP) and smoothing the project switching costs. In the operations literature, smoothing is often seen as an inherently desirable goal, for example, Graves [1986] set up an objective function to emphasize production smoothing. Eq. (3.3) quadratically weights the difference between actual WIP and a nonnegative target. The form of this penalty is exactly analogous to that of the HMMS model [Holt et al. 1960] or Sethi and Thompson's [2000] penalty for inventory fluctuations. As in both models the WIP target is typically greater than zero, albeit often small in the models, the penalty for undershooting the WIP "target" exists to avoid idling workers through WIP starvation. Somewhat more complex yet analogous issues apply to WIP in professional services. Because the opportunity cost of lost revenue relative to salary costs is on the order of a factor of twenty, generally development firms consider zero-WIP (which implies idling capacity) unacceptable. For these reasons, and because all the aforementioned costs are convex, we follow literature in penalizing deviations from "target" WIP levels in a quadratic manner [Mihm et al. 2003].

For brevity, we assume that target WIP levels are determined prior to assessing the penalty on deviation from targets. The cost of switching projects is also quadratic weighted, as is typical in the capacity planning literature (e.g., Holt et al. [1960] and Sethi and Thompson [2000]). Moving projects from one site to another often involves some sort of setup cost [Anderson and Joglekar 2005]. Further, transferring five projects per month from the principal to a partner is logically much simpler than fifty. Hence, the coordination problem tends to escalate in a convex manner, suggesting the quadratic form. In summary, the objective function is designed to balance project switching costs against service capacity utilization at both principal and partner.

3.4. Control with Inaccurate Data

The model in Figure 3 builds on Figure 2 by allowing inaccuracies. The data are filtered to control the outsourcing decision using a control law derived assuming perfect data. The managers keep aggregate statistics (means and variance) on the estimation and measurement errors. These measurement errors are assumed to be independent, white noise processes.

3.5. Analysis—The Optimal Control Law

We derive the optimal control law based on the objective function in Eq. (3.3). We then select the filter characteristics that provide minimal cost penalty. For convenience, let

$$\mathbf{x}(t) = \begin{bmatrix} x_i(t) \\ x_o(t) \end{bmatrix}, \mathbf{w}(t) = \begin{bmatrix} w_i(t) \\ w_o(t) \end{bmatrix}, \mathbf{z}(t) = \begin{bmatrix} z_i(t) \\ z_o(t) \end{bmatrix}, \mathbf{n}(t) = \begin{bmatrix} n_i(t) \\ n_o(t) \end{bmatrix}, \tag{4.1}$$

and

$$\mathbf{F} = \begin{bmatrix} 0 & \phi\lambda \\ 0 & -\lambda \end{bmatrix}, \mathbf{G} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} \beta_i^2 & 0 \\ 0 & \beta_o^2 \end{bmatrix}, \mathbf{R} = [1], \mathbf{W} = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_o^2 \end{bmatrix}, \mathbf{N} = \begin{bmatrix} \nu_i^2 & 0 \\ 0 & \nu_o^2 \end{bmatrix}. \tag{4.2}$$

Recall that the vectors $\boldsymbol{x}(t)$, $\boldsymbol{w}(t)$, $\boldsymbol{z}$, (t), and $\boldsymbol{n}(t)$ represent WIP, variation in WIP, WIP that is observed and has quality errors, and the amount of noise in the WIP because

of quality errors at a given time $\boldsymbol{t}$. Here $\boldsymbol{F}$, $\boldsymbol{G}$, $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{W}$, and $\boldsymbol{N}$ represent the matrices for completion-rate fractions, direction of their assignment, cost penalty, normalizing constant, the variance in $\boldsymbol{w}(\boldsymbol{t})$ defined earlier, and the variance in $\boldsymbol{n}(\boldsymbol{t})$ defined earlier. Accordingly, $\boldsymbol{n_i}(\boldsymbol{t})$ = the noise in the estimation at the principal firm. $\boldsymbol{n_i}(\boldsymbol{t})\boldsymbol{N}(\boldsymbol{0}, \upsilon_i^2)$ and $\text{corr}(\boldsymbol{n_i}(\boldsymbol{t})), \boldsymbol{n_i}(\boldsymbol{t}+\boldsymbol{\varepsilon})) = \boldsymbol{0}$, for $\boldsymbol{\varepsilon} \neq \boldsymbol{0}$. Moreover, $n_o(t)$ is the noise in the data at the partner site. $n_o(t)$ $N(0, \sim \upsilon_o^2)$ and $\text{corr}(n_o(t), n_o(t+\varepsilon)) = 0$, for $\varepsilon \neq 0$. We write the evolution of the state variable, that is, number of in-sourced and outsourced projects, as.

$$\begin{bmatrix} x_i(t) \\ x_o(t) \end{bmatrix} = \mathbf{F}[\mathbf{x}(t) - \tilde{\mathbf{x}}(t)] + \mathbf{G}u(t) + \mathbf{w}(t). \tag{4.3}$$

If the information is imperfect due to measurement error, the observed state variables are z(t).

$$\mathbf{z}(t) = \mathbf{x}(t) + \mathbf{n}(t). \tag{4.3b}$$

THEOREM 1. *The optimal control policy $u^*(t)$ with perfect measurement (perfect data) is given by*

$$u^*(t) = \mathbf{k}_c(\mathbf{x}(t) - \tilde{\mathbf{x}}) \text{ where } k_c = \left[\beta_i + \lambda - \frac{\beta_i}{\sqrt{\beta_o^2 + \beta_i^2 + \lambda^2 + 2\lambda\beta_i(1-\phi)}}\right]^T = [k_{c,in}\ k_{c,out}]. \tag{4.4}$$

Superscript T represents a transpose operation.

An important managerial test that this policy must pass is whether it results in a "stable" system, that is, neither of the WIPs can "run away" by growing without bound towards infinity under a finite demand. Such behavior is clearly undesirable. We would also like to determine whether or not the optimal policy creates oscillations over and above the disturbances inherent in the demand and reintegration rates themselves. Smith and Eppinger [1997] tested for this in their project WIP model by borrowing the concept of "overdamping" from control theory. We will use the term "damping" to indicate that the outcome from a data filter attenuates the inputeffects. Specifically, "overdamped" indicates that a system's state variables do not overshoot their final values in response to a one-time increase in demand, whereas "underdamped" indicates that they do [Oppenheim et al. 1983, page 245]. Hence, because any oscillations from the input disturbances are attenuated, the state variables in an overdamped system will be, in some sense, "smoother" than the input, whereas in an underdamped system, they will be "rougher." We now seek to determine whether the optimal policy creates a stable system and, if so, whether it is over-or underdamped.

COROLLARY 1. *The two eigenvalues for the system using the optimal control policy $u^*(t)$ are*

$$\tfrac{1}{2}\left[-\sqrt{\beta_o^2 + (\beta_i + \lambda)^2 - 2\beta_i\lambda\phi} \pm \sqrt{\beta_o^2 + (\beta_i - \lambda)^2 + 2\beta_i\lambda\phi}\right]. \tag{4.4b}$$

PROPOSITION 1. *The state variables under the optimal control policy are stable, that is, any bounded demand will generate bounded completion rates. Further, the control policy will create an overdamped system. Thus, run-away project WIPs are impossible under the optimal policy and furthermore the oscillation of the two WIP is, in some sense, strictly a result of demand and reintegration disturbances. The optimal policy does not increase this oscillation.*

PROPOSITION 2. *The magnitude of the control law weight on in-sourced projects increases in the cost β_i of deviating from the target in-sourced projects. The magnitude of*

the control law weight on outsourced projects increases in the cost β_i of deviating from the target in-sourced projects and the rate at which outsourced projects are completed λ. It decreases in cost β_o of exceeding the target outsourced projects and reintegration fraction $\phi$of outsourced projects.

THEOREM 2. *Evolution of minimum mean square estimate of $\boldsymbol{x}(t)$ is governed by a Kalman-Bucy filter*

$$\hat{\mathbf{x}}'(t) = \mathbf{F}[\hat{\mathbf{x}}(t) - \tilde{\mathbf{x}}(t)] + \mathbf{G}u(t) + \mathbf{K}_e[\mathbf{z}(t) - \hat{\mathbf{x}}(t)]. \tag{4.5}$$

Recall that we have already explained $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{z}$ terms of this equation earlier. The notation ***x-hat*** implies that it is an estimate. The prime represents the first derivative with respect to time ($\boldsymbol{t}$). Here $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{P}$ represent the matrices for completion-rate fractions, direction of their assignments, and a normalization matrix. $\boldsymbol{K}$ is the matrix of Kalman-Bucy filter weights applied to the noisy data as shown in Corollary 2 shortly.

$$\text{Moreover, } \mathbf{K}_e = \mathbf{P}\mathbf{N}^{-1}\text{such that } \mathbf{P}\mathbf{N}^{-1}\mathbf{P} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T + \mathbf{W} \text{ and } \mathbf{P} \text{ is positive definite.} \tag{4.6}$$

COROLLARY 2. *If the fraction of reintegration work as a result of outsourcing $\phi = 0$, then*

$$\mathbf{K}_e = \begin{bmatrix} \dfrac{\sigma_i}{\nu_i} & 0 \\ 0 & -\lambda + \sqrt{\lambda^2 + \dfrac{\sigma_o^2}{\nu_o^2}} \end{bmatrix}. \tag{4.7}$$

COROLLARY 3. *Assume that the fractional rate of outsourced project reintegration $\phi = 0$. Then the four Eigenvalues for the system using the optimal control policy $u^*(t)$ in Corollary 2 are*

$$\frac{1}{2}\left[-\sqrt{\beta_o^2 + (\beta_i + \lambda)^2} \pm \sqrt{\beta_o^2 + (\beta_i - \lambda)^2}\right], -\frac{\sigma_i}{\nu_i} \text{and} -\sqrt{\frac{\sigma_o^2}{\nu_o^2} + \lambda^2}. \tag{4.8}$$

PROPOSITION 3. *Assume that the fractional rate of outsourced project reintegration, $\phi = 0$. Then, the rate at which the estimate of in-sourced projects $\hat{x}_i(t)$ is updated increases in the variance of the demand and in-sourced completion rates σ_i and decreases in the variance of the measurement error of in-sourced projects ν_i. The rate at which the estimate of outsourced projects $\hat{x}_o(t)$ is updated increases in the variance of the outsourced completion rates σ_o and decreases in the fractional outsourced WIP completion rate λ and the variance of measurement error of in-sourced projects ν_i.*

PROPOSITION 4. *Assume that the fractional rate of outsourced project reintegration $\phi = 0$. Under imperfect information, the state variables under a combination of the optimal control policy and the optimal Kalman-Bucy filter are stable. That is, any bounded demand will generate bounded completion rates. Furthermore, the control policy is guaranteed to create an overdamped system only when the reintegration work fraction $\phi = 0$.*

(Note: from numerical studies, if the reintegration work fraction $\phi > 0$, then the system may be underdamped.) Hence, there can be no "run-away" portfolio of projects at an aggregate level even with imperfect information. However, if outsourced work requires integration, the WIPS may no longer necessarily be smoother than their input disturbances.

THEOREM 3. *The penalty function for the case with perfect information is*

$$J_p = \mathbf{S}\mathbf{W}. \tag{4.9}$$

where

$$\mathbf{S} = \frac{1}{\lambda(\phi-1)}\begin{bmatrix} \beta_i(\beta_i-\tilde{\beta}) & \beta_i\left[\beta_i+\lambda(1-\phi)-\tilde{\beta}\right] \\ \beta_i\left[\beta_i+\lambda(1-\phi)-\tilde{\beta}\right] & (\beta_i+\lambda)^2-\lambda\phi(2\beta_i+\lambda)-\left[\beta_i+\lambda(1-\phi)\right]\tilde{\beta} \end{bmatrix} \tag{4.9a}$$

and where $\tilde{\beta} = \sqrt{\beta_o^2+\beta_i^2+\lambda^2+2\lambda\beta_i(1-\phi)}$.

S, in Eq. (4.9a), is a matrix that consists of 3 cost terms. The term on the top left is the penalty incurred at the principal. The term on the right bottom corner is the penalty cost incurred by the partners. The off-diagonal terms are the penalty costs associated with transfers between principal and partners.

The penalty function for the case with imperfect information under the Kalman-Bucy filter and control is

$$J_k = J_{ce} + J_s \text{ where } J_{ce} = \mathbf{SK}_e(\mathbf{N}+\mathbf{P})\mathbf{K}_e^T \text{ and } J_s = \mathbf{QP}. \tag{4.10}$$

The penalty function has two parts. The first term J_{ce} is the term S described earlier, but corrected for the variances based on certainty equivalence in the process because of the noisy data. The second term is the steady-state cost penalty (recall that P and K are explained before). The penalty function for the case with imperfect information (and $\phi = 0$) when the control law $u^*(t)$ is used directly upon the corrupted measurements **z** as if they were perfect measurements of the state variables **x** is

$$J_m = J_p + J_n, \text{ where } J_n = \frac{\left[(k_{ci}-k_{co})^2+k_{ci}\lambda\right]\left(k_{ci}^2v_i^2+k_{co}^2v_o^2\right)}{2(k_{ci}-k_{co}+\lambda)}. \tag{4.11}$$

4. MODEL INSTANTIATION: NUMERICAL ANALYSES

Our objective in instantiating the model is to illustrate how the model may be applied in practice. To make this case, we explore the behavior of the system in response to a range of plausible inputs. We present a series of numerical analyses using the control law and Kalman-Bucy filter derived earlier. The performance with perfect data is discussed first and that with inaccurate data later. The results from perfect data serve as a benchmark for exploring the efficacy of the filter.

4.1. Benchmark Scenario: Performance with Perfectly Accurate Data

Numerical studies, while assuming perfect data quality, provide a benchmark for studying the effects of inaccurate data presented in Section 4.2. The response of the system with asymmetric costs at the principal and partners is explored initially, and the response of the system to changes in the integration work fraction ϕ and the project completion rate at partners λ is presented afterwards. (For ease of exposition, these costs are assumed symmetric in the second set of studies.) Three types of performance measures are included in both these studies:

(i) the optimal control policy $u(t)$ coefficients of $\mathbf{k_c}$ on in-sourced and outsourced WIP, which for the ease of description shall be termed as $k_{c,in}$ and $k_{c,out}$, respectively;
(ii) decay rates (δ_1 *and* δ_2) for the natural response of the system (these values are computed based the Eigen-values in Eq. (4.4b); and
(iii) the penalty function for the case with perfect data J_p (based on Eq. (4.9).

Table I illustrates the impact of an asymmetric cost structure, assuming that the integration work fractions ($\phi = 0.5$) and the project completion rate at partners ($\lambda = 1$) are constant. Standard deviation of disturbances to the demand and completion rates at each stage are $\sigma_i = 1$, $\sigma_o = 1$. The variation in the values of the control law coefficient $k_{c,out}$ is logical given the model. When the principal's internal WIP deviation penalty

Table I. Impact of Cost Structure on Optimal Assignment

β_i	β_o	$k_{c,in}$	$k_{c,out}$	δ_1	δ_2	J_p
1.00	**0.10**	−1.00	0.63	0.36	1.37	1.50
1.00	**1.00**	−1.00	0.00	0.29	1.71	2.44
1.00	**10.00**	−1.00	−0.85	0.05	10.10	31.01

Table II. Impact of Partner Characteristics on Optimal Assignment

λ	ϕ	$k_{c,in}$	$k_{c,out}$	δ_1	Δ_2	J_p
0.10	**0.00**	−1.00	0.39	0.07	1.42	9.18
1.00	**0.00**	−1.00	0.24	0.62	1.62	1.71
10.00	**0.00**	−1.00	0.05	0.99	10.05	1.05
0.10	**0.50**	−1.00	0.35	0.04	1.42	17.5
1.00	**0.50**	−1.00	0.00	0.29	1.71	3.00
10.00	**0.50**	−1.00	−0.42	0.50	10.09	2.42

is larger than the partner's ($\beta_i > \beta_o$), a larger fraction of projects from the principal will be moved to the partner (i.e., $k_{c,in} < k_{c,out}$), transferring more WIP oscillation to the partner. In contrast, when the internal WIP deviation penalty is smaller than the partner's, the reverse is true. We conjecture that the first case ($k_{c,in} < k_{c,out}$) will prevail in practice because most organizations seem to outsource in order to minimize variation in the utilization of their own resources. We include the second for completeness. We note that the two coefficients, based on our choice of control parameters ($\phi = 0.5$ and $\lambda = 1$), show some asymmetry.

The decay rates δ_1 and δ_2 are measures of system stability. Run-away work in process occurs whenever either value of δ is negative. The values of J_p in the table confirm Proposition 1 in that the positive real values of all the decay rates in a system indicate that it is both stable and overdamped, that is, for these parameter sets, the optimal policy will create neither run-away WIPs nor increase the oscillation of the WIPs above that of demand. Finally, consistent with modeling intuition, outsourcing reduces the overall cost penalty J_p when the costs of deviating from WIP targets are more at the principal than at the partner.

In the next set of tests, the effect of partner characteristics is explored by assuming that the cost structure is symmetric and constant as are the disturbances ($\beta_i = \beta_o = 1.0; \sigma_i = \sigma_o = 1$), while the integration work fraction ϕ and the project completion rate at partners λ vary in a systematic manner as shown in Table II. The numerical values of the control law coefficients, $k_{c,in}$ and $k_{c,out}$ behave reasonably. When the fractional project completion rate at partner, λ, increases, more oscillation in work is transferred to the partners. Similarly, when the reintegration penalty ϕ rises, less oscillation in work is transferred to the partner. Again, all the δ values are positive, which corroborates that the system is overdamped and stable. Finally, the project completion rate at partners λ reduces the overall penalty cost J_p, and the reintegration penalty ϕ increases it. We have not shown interaction effects between these tables in the interest of saving space. However, data from Tables I and II should suffice to show that the system cost structure and the characteristics of partners affect performance measures. In the next sections we will assume that the cost and system characteristics are invariant. Appropriate values of overall penalty costs J_p will be used as a benchmark for studying the performance with imperfect information.

4.2. Performance with Inaccurate Data

Figure 4 illustrates the system behavior with inaccurate data, considering two policies. One is to use the optimal control law from Theorem 1 assuming that the measurement

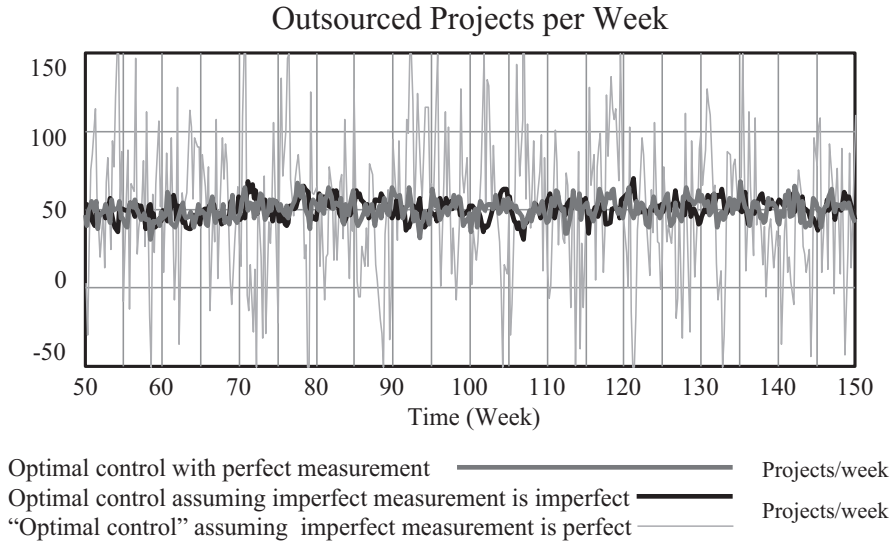

Fig. 4. Evolution of outsourcing policy due to variation in incoming projects $\beta_\iota = 1$, $\beta_o = 2$, $\sigma_\iota = \sigma_o = 10$, $\nu_\iota = \nu_o = 20$, $\phi = 0$, and $\lambda = 0.2$.

Table III. Performance under Varying Amounts of Noise

ν_i	ν_o	$\mathbf{J_p}$	$\mathbf{J_m}$	$\mathbf{J_{CE}}$	$\mathbf{J_S}$	$\mathbf{J_K}$
0.01	**0.01**	1.71	1.71	171.12	0.02	171.14
0.10	**0.10**	1.71	1.72	17.48	0.19	17.63
0.10	**1.00**	1.71	1.79	13.71	0.51	14.22
1.00	**0.10**	1.71	2.94	6.36	1.09	7.45
1.00	**1.00**	1.71	3.01	2.59	1.41	4.00
1.00	**10.00**	1.71	9.83	2.47	1.00	3.97
10.00	**1.00**	1.71	125.38	1.50	10.00	11.89
10.10	**10.00**	1.71	132.20	1.36	10.52	11.86

is perfect, even though it is not. The other is to assume the data are inaccurate and use a filter, a Kalman-Bucy filter, to obtain more accurate data on the number of projects in the in-sourced and outsourced WIP stocks. Notice that the deployment of the optimal control policy, using the filtered data, results in a response that closely tracks the system response with perfect data.

On the other hand, using the optimal control law without filtering the imperfect data leads to a much larger variance in the number of outsourced projects per week. The results are consistent with intuition; absence of erroneous estimates when using an optimal policy attenuates project switching, because the error terms are denominators in the filter multipliers (see Eq. (4.7). In Table III, we explore the contribution of individual system parameters to this attenuation process. Control parameters for these simulations have been set at $\beta_i = \beta_o = 1, wi = wo = 1, \phi = 0$, and $\lambda = 1$. Three performance measures are explored for varying levels of measurement (ν_i and ν_o):

(i) $\boldsymbol{J}_p$: penalty function for the case with perfect data;
(ii) $\boldsymbol{J}_m$: penalty for optimal control assuming that the inaccurate data are perfect;
(iii) $\boldsymbol{J}_k$: penalty for optimal control assuming that the inaccurate data are imperfect and hence have filtered before use. There are two components of this penalty: $J_k = J_{CE} + J_S$. Here J_{CE} is the penalty associated with the estimate error and J_S is the penalty associated with the measurement error.

These results show that inaccurate data always creates a performance penalty. In the most benign conditions (for high signal-to-noise ratios, $\omega/\nu \to 0$), and assuming that the optimal control policy is known and but measurements are not filtered, the performance degradation from J_p to J_m is not significant. As the error level increases, however, the performance of the system without using the filter gets progressively worse.

When the variances in measurement errors are much smaller than variations in random disturbances to demand and to project-completion rates, the Kalman-Bucy filter results in a decidedly worse performance, J_k. It improves rapidly and becomes very significant as the measurement errors increase. The behavior of Kalman-Bucy-type data filters, depending on system characteristics, has been known to exhibit singularities at very high signal-to-noise ratios. In such situations enhancements to the derived filter have been advocated. On the other hand, even the current filter can improve performance in noisy environments. Further, these data show an interesting trade-off between J_{ce} versus J_s as function of the "signal-to-noise" ratio ω/ν. Recall that J_s corresponds to the penalty associated with the measurement error, as shown by the covariation term $\boldsymbol{P}$ in Eq. (4.10). J_{ce}, on the other hand, is linked with the certainty equivalence, and its value is dependent on the filter gain matrix $\mathbf{K_e}$. Eq. (4.7) confirms that the filter gains $\mathbf{K_e}$ are inversely related to the noise level. Hence, the J_{ce}-J_s trade-off can be explained as follows: at high signal-to-noise ratio, the penalty due to J_s is minimal; however, because the signal drowns out the noise and the filter gains are very large, the system becomes highly sensitive to small perturbations. With a low signal-to-noise ratio, in contrast, the measurement errors create the need for more adjustments and consequently increase the overall cost penalty.

5. DISCUSSION

The economic incentive associated with improving quality of project-status data is evident, as shown by the growth of J values with higher noise levels in Table III. We now discuss the challenges with implementing our results by first presenting the extant practices that lead to data quality problems, particularly in inter-organizational project settings. We then examine timeliness and how the consumption of real-time data is supported by the filtering solution proposed. We finally comment on how to characterize the noise associated with aggregate project-status data and describe how the model data and model parameters may be obtained to implement our solution in practice. The model analyzed in this article deals with a principal and several partners who are aggregated. The model and findings can be extended without loss of generality to multiple partners, each with its own capacity and integration burden.

5.1. Sources of Inaccuracy

There are three potential sources of errors in data associated with distributed projects: technical limitations within information processing systems, data collection errors, and gaming.

—*Information System Errors*. Information systems are not error free. Only a fraction of system errors can be attributed to algorithmic or computational errors. A much larger fraction of the errors may be due to exchanging data across multiple systems, combining data from multiple sources, and from legacy applications. These factors have been identified in the literature on data quality and methods proposed to ameliorate the impact of these errors [Redman 1996; Wang 1998; Lee et al. 2006]. However, in distributed project settings, these errors cannot be completely eliminated and managers need to deal with inaccurate data. The solution we propose offers the

ability to smooth data errors so that the managers may have more accurate data. We have shown (by comparing J_m *and* J_K in Table III) that this improves outsourcing decisions and allows for decisions that are economically superior.

—*Data Collection Errors*. The second set of concerns revolves around an organization's ability to collect project-status data in a timely manner. Some attention has been paid in the information systems literature, for instance, the need to understand the value of information and frequency of updates in a supply chain management context [DeHoratius 2004]. This leads to project-status data being collected only on a periodic basis and their aggregation remains a point of concern [Anderson and Joglekar 2005]. It is quite difficult to nail down the amount of measurement error generated by such processes. Moreover, in some instances, the distributed nature of product development work creates what has been described as "project oscillatory behavior" [Mihm et al. 2003; Yassine et al. 2003] over and above what has been discussed in this article. Such oscillations can compound the data quality problems. The proposed filtering solution smoothens a significant number of these errors. Further, the solution allows for managers to incorporate real-time data into their outsourcing decision. The incorporation of real-time data into the decision process is discussed further in Section 5.3.

—*Gaming*. It is not a surprise that a big concern for data quality is the possibility of gaming of project-status data. Ford and Sterman [2003] report that project-status data are routinely misreported under what they term as the "liar's poker syndrome" wherein the nature of reintegration burdens makes it rational for managers to provide erroneous data. Of course, savvy project managers typically adjust for this by discounting such reports by a fixed "fudge factor." However, the likelihood that the "fudge factor" exactly balances out underreporting at every reporting period remains low. There is not a lot that our solution can do about gaming. Contracts between the principal and partners may be strengthened and incentive schemes that motivate partners to minimize gaming included. If a partner's scheme for gaming is consistent, the filter can adjust for this noise and smooth the aggregate data. In the next section, we comment on how the estimates of errors can be established and used to fine-tune filters to improve quality.

5.2. Obtaining Model Parameters

A practical question that arises in anticipation of using our model is: how should a manager obtain the data and estimate parameters to set up the control law and the filter? Table IV lists all model parameters that need to be estimated and the source of data for obtaining each estimate. Using the named parameters given earlier and Eqs. (4.7) through (4.10), a manager can design the optimal filter and compute the penalty costs as shown in illustrative examples summarized in Tables I and II. Further, a manager can verify the impact of "fudging" or approximating the noise by generating time series that go with the illustrative example shown in Figure 4. It is evident that there is a significant difference between the use of a Kalman-Bucy filter and without it. Corresponding cost savings are computed and shown in Table III. In addition, a key managerial lesson learned from the mathematical work is that the Kalman-Bucy filter is designed by understanding the noise of the incoming data and the measurement processes. In our model we have assumed the noise to be stationary or the variance of the noise as constant. One of the key jobs of the manager is to constantly check the noise and its variances and adjust the filter as the variances in noise shift. A related lesson of practical importance is that if the noise level falls to low levels, recall from the discussion that follows Table III that the performance of the Kalman-Bucy filter declines considerably and may not be cost effective to implement.

Table IV. Estimated Model Parameters and Sources

Model Parameter	Parameter Description	How Obtained?
c	Completion rate at the principal site	Based on capacity at the principal site - measured in terms of lines of code or modules per person-hour.
λ	Fractional completion rate of WIP per unit time at the partner	Based on capacity at the partner site - ask the partners for the historical data.
$w_i(t)$	Net variation in stock of WIP at the principal, $w_i(t) \sim N(0, \sigma_i^2)$	Parameter estimated from historical data at the principal - e.g. estimate standard deviation σ_i of time series in Figure 1.
$w_o(t)$	Net variation in stock of WIP at the partner, $w_o(t) \sim N(0, \sigma_o^2)$	Parameter estimated from historical data at the partner site. E.g. estimate standard deviation σ_o of time series in Figure 1.
$n_i(t)$	The measurement noise in the variation in the changes in stock of WIP at the partner, $n_i(t) \sim N(0, \nu_i^2)$	Standard deviation parameter can be estimated by historical data at the principal - e.g. based on the gap between estimated and observed time series in Figure 1.
$n_o(t)$	The measurement noise in the variation in the changes in stock of WIP at the partner, $n_o(t) \sim N(0, \nu_o^2)$	Standard deviation parameter can be estimated by historical data from the partner - e.g. based on the gap between estimated and observed time series of time series in Figure 1.
$\beta_i^2(t)$	The cost penalty for in-sourced backlog	Parameter is determined by the contract between the principal and the consumer (and updated based on historical data).
$\beta_o^2(t)$	The cost penalty for out-sourced backlog	Parameter is determined by the contract between the principal and the partner (and updated based on historical data).

5.3. Incorporation of Real-Time Data

It is important to note that we have not treated timeliness as a contributing factor to the overall accuracy of project-status data, that is, we have only considered accuracy and we have assumed timeliness to be subsumed by accuracy in this context. Literature has discussed the interrelationship between accuracy and timeliness [Ballou and Pazer 1995] and has emphasized timeliness as a key data quality dimension (e.g., Pipino et al. [2002], Lee et al. [2006], and Madnick et al. [2009]). While techniques to measure timeliness have been proposed, methodologies to improve timeliness, besides the use of sensors and sensory networks that offer real-time data (e.g., Gaynor and Shankaranarayanan [2008]) have not been addressed. A plausible explanation for this methodological gap, in the context of managing distributed projects, is the difficulty with collecting project-status data in a timely manner. The data that serves as the input to the control law and filter proposed in this article are shown in Table V.

Literature has also argued that timeliness is a contextual dimension and depends on the decision context [Pipino et al. 2002; Fisher et al. 2003; Lee et al. 2006]. Therefore, when a manager uses the latest project-status data (i.e., data from the most recent updates), in the context of outsourcing decisions in distributed projects, the updates may be considered timely and even real time. When aggregating data across multiple projects and partners, if a subset of the data needed for aggregation is updated recently and another subset is from a previous update (because that subset was not collected in a timely manner), the resulting aggregate data are more inaccurate as compared with data aggregated when all of the aggregated data are based on the most recent update. When an organization is unable to collect all of the project-status data required for aggregation in a timely manner, managers are unable to use the most updated data elements in the decision process, even when these are available.

Table V. Model Data and Sources

Model data	Description of the data	How obtained?
$r(t)$	End-customer demand rate	This is the rate at which the end customer wants projects delivered. This is a real-time data input.
$x_i(t)$	the work in progress, measured in number of hours of work, at the	Real time data generated at the principal
$x_o(t)$	the work in progress, measured in number of hours of work, at the	Real time data generated at the partner
λ	The fraction of tasks completed at the partner requiring re-integration at the principal ($o < \lambda < 1$)	Real time data generated based on the tasks that comes back to the principal for integration
$\tilde{x}_i(t)$	Targeted insourced work in progress	This is target value for the work insourced by the principal. In this paper, we assume that the principal sets this target. Therefore it is data obtained from the principal. Theoretically, this value can be optimized to maximize capacity and utilization. For the dataset in Figure 1, this is the mean of the time series.
$\tilde{x}_o(t)$	Targeted outsourced work in progress	This is target value for the work outsourced to partners by the principal. In this paper, we assume that the principal sets this target. Therefore it is data obtained from the principal. Theoretically, this value can be optimized to maximize capacity and utilization. For the dataset in Figure 1, this is the mean of the time series.

Recall from the literature that collaborative filtering [Sahoo et al. 2011] provides mechanisms that aggregate data from multiple sources and offer it together, thereby avoiding the time-lag that occurs when one tries to get data individually from multiple sources. We posit that the application of the filter offers managers the ability to consume the most recently updated data, even if this is only a subset of the data required for aggregation. The filter "smoothes" the errors to a certain extent and is more effective when the magnitudes of errors are large.

5.4. Inter-Organizational Data Exchange

Given these sources for errors in project-status data and the difficulty with correcting these errors early on, managers have to make outsourcing decisions despite their awareness about project-status data inaccuracies. Our research offers an initial step towards improving the accuracy of the data. Further, in distributed project settings, organizations are dealing with data that comes from another different organization. Literature has argued that in order for an organization to be assured that they are receiving good-quality data from another organization, the two organizations must agree upon standards and operate within these quality standards. Alternately, they must cooperate by sharing not only the quality metadata (i.e., quality measurements of the data that is exchanged), but also the standards for how these measurements were derived [Cai and Shankaranarayanan 2007]. The solution for improving accuracy proposed here offers another method for ensuring quality (specifically, accuracy) of incoming data. If the sending organization consents to sharing related historical data, the receiving organization can estimate the errors (deviations).

Collecting such historical data (project-status data) is not a trivial exercise. Browning et al. [2002] discuss several reasons in the context of developing an Uninhibited Combat Aircraft Vehicle (UCAV) to illustrate the lack of precision in monitoring the progress. We account for these difficulties while selecting key parameters that are needed to set

up our model: the costs structure parameters β_i and β_o, and the completion rate at the partner site (λ). Users may estimate these parameters using a regression model adopted by Morrice et al. [2004]. Similarly, the noise (error) associated with the progress data can be estimated by mining project-status data [Braha 2001]. Managers are aware of the inaccuracies in project-status data and have an incentive to write a contract requiring the partners to share historical data. The reader will recall from the sensitivity analyses presented in Table III that the performance penalties associated with poor data quality rise with the level of noise in the project-status data.

A key issue with the application of filters is that it is not easy to obtain stable measurements of signal-to-noise ratios within a single project. Hence, aggregate tracking of data across multiple projects has been used to demonstrate and instantiate the use of filters to improve data quality. The methodology is also applicable when data are exchanged within an organization, between business units, such as the case when projects are in-sourced.

6. CONCLUSION

In this article, we develop a solution for managing the quality of the data exchanged between organizations when managing distributed projects. In the presence of inaccuracies in project-status data, we explore the impact of cost structure asymmetry on optimal division of projects between the principal and its partners. Our analysis along with numerical instantiation in Figure 4 shows that with perfect data, such a system will not create run-away project-WIP and, in fact, will smooth it. Further, in Table III, we determine the value of obtaining perfect project-status data and the value of filtering when data are inaccurate.

Our research provides a method to quantitatively measure the benefits of improving accuracy of project-status data in the context of sourcing decisions. In particular, we offer case-based evidence together with model-based analysis, arguing that managing the quality of aggregate project-status data can be a serious issue in many firms. There are several limitations to this work. One is our choice of a quadratic cost structure, which yields a linear control rule. Some managers may want to run their operations under linear control because such rules are easy to understand. However, such rules are typically not used for systems with extremely variable demand or significant peak capacity utilization or idling. Another concern is the performance of the filter under conditions of a high signal-to-noise ratio. We have assumed, in this article, that the noise variable is stationary. This allows us to estimate the value of the variable to set the filter. A third issue is what if the noise variable is not stationary? This would require us to constantly monitor the variable, obtain the changing values, and dynamically adjust the filter. This clearly will increase the costs associated with the filtering solution and change the cost structure and associated economics. These issues have a direct bearing on how much one should pay, either within the principal or to the partner, for a metering process to reduce the level of noise in the project-status data below a specified threshold. These issues suggest that managing data in distributed, and typically inter-organizational, settings offers a rich frontier for further research in data and information quality.

ELECTRONIC APPENDIX

The electronic appendix to this article is available in the ACM digital library.

REFERENCES

ANDERSON, E. G. AND JOGLEKAR, N. R. 2005. A hierarchical product development planning framework. *Prod. Oper. Manag. 14*, 2.

Ballou, D. P., Wang, R. Y., Pazer, H., and Tayi, G. K. 1998. Modeling information manufacturing systems to determine information product quality. *Manag. Sci. 44*, 4, 462–484.

Ballou, D. P. and Pazer, H. L. 1995. Designing information systems to optimize the accuracy-timeliness tradeoff. *Inf. Syst. Res. 6,* 1, 51–72.

Banker, R. D. and Kaufmann, R. J. 2004. The evolution of research on information systems: A fiftieth-year survey of the literature in management science. *Manag. Sci. 50*, 3, 281–289.

Bertsekas, D. 2001. *Dynamic Programming and Optimal Control 2nd Ed.* Athena Scientific, Belmont, MA.

Braha, D. 2001. *Data Mining for Design and Manufacturing: Methods and Applications.* Kluwer Academic.

Browning, T. R., Devst, J. J., Eppinger, S. D., and Whitney, D. E. 2002. Adding value in product development by creating information and reducing risk. *IEEE Trans. Engin. Manag. 49*, 4, 443–458.

Caballero, I., Vizcaíno, A., and Piattini, M. 2009. Optimal data quality in project management for global software developments. In *Proceedings of the 4th International Conference on Cooperation and Promotion of Information Resources in Science and Technology.*

Cai, Y. and Shankaranarayanan, G. 2007. Managing data quality in inter-organizational data networks. *Int. J. Inf. Qual. 1*, 3, 254–271.

Carrascosa, M., Eppinger, S. D., and Whitney, D. E. 1998. Using the design structure matrix to estimate time to market in a product development process. In *Proceedings of the ASME Design Theory and Methodology Conference.* 98–6013.

Chen, Y. and Bharadwaj, A. 2009. An empirical analysis of contract structures in IT outsourcing. *Inf. Syst. Res. 20*, 4, 484–506.

Chiang, R. I. and Mookerjee, V. 2004. A fault threshold policy to manage software development projects. *Inf. Syst. Res. 15*, 1, 3–21.

Cusumano, M. A. and Selby, R. W. 1995. *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. Free Press, New York.

Dehoratius, N. 2004. In pursuit of information quality. *Cutter IT J. 17,* 9, 2–4.

Dominguez, A. 2006. Project management in noisy environments. In *Proceedings of the POM Annual Meeting*.

Even, A. and Shankaranarayanan, G. 2007. Utility driven assessment of data quality. *Datab. Adv. Inf. Syst. 38*, 2, 75–93.

Fisher, C. W., Chengalur-Smith, I., and Ballou, D. P. 2003. The impact of experience and time on the use of data quality information in decision-making. *Inf. Syst. Res. 14*, 2, 170–188.

Ford, D. and Sterman, J. 2003. The liar's club: Impacts of concealment in concurrent development projects. *Concurr. Engin. Res. Appl. 11*.

Gaynor, M. and Shankaranarayanan, G. 2008. Implications of sensors and sensor-networks for data quality management. *Int. J. Inf. Qual. 2*, 1, 75–93.

Gomes, P. J. and Joglekar, N. R. 2008 Linking modularity with problem solving and coordination efforts. *Manag. Decis. Econ. 29*, 5, 443–457.

Graves, S. 1986. A tactical planning model for a job shop. *Oper. Res. 34*, 522–533.

Hernadez, M. A. and Stolfo, S. J. 1998. Real-world data is dirty: Data cleansing and the merge/purge problem. *J. Data Mining Knowl. Discov. 1,* 2.

Hirschheim, R., Heinzl, A., and Diebern, J. 2002. *Information Systems Outsourcing.* Springer.

Holt, C. C., Modigliani, F., Muth, J. F., and Simon, H. A. 1960. *Planning Production, Inventories, and Work Force.* Prentice-Hall, Englewood Cliffs, NJ.

Jones, M. C. and Price, R. L. 2004. Organizational knowledge sharing in erp implementation: Lessons from industry. *J. Organ. End User Comput. 16*, 1, 21–41.

Jorgensen, M. 2007. Forecasting of software development work effort: Evidence on expert judgment and formal models. *Int. J. Forecast. 23*, 3, 449–462.

Lee, Y. W., Pipino, L. L., Funk, J. D., and Wang, R. Y. 2006. *Journey to Data Quality.* MIT Press, Cambridge, MA.

Madnick, S., Wang, R. Y., and Lee, Y. W. 2009. Overview and framework for data and information quality research. *ACM J. Data Inf. Data Qual. 1*, 1–22.

Makridakis, S. and Winkler, R. 1983. Averages of forecasts: Some empirical results. *Manag. Sci. 29*, 987–996.

Mandel, M. and Engardio, P. 2007. The real cost of off-shoring. *BusinessWeek,* June 18.

Mcgrath, M. 2001. How to boost r&d productivity by 50%? *Insights Mag.* www.prtm.com

Meredith, J. R. and Mantel, S. J. 2005. *Project Management: A Managerial Approach.* Wiley, New York.

Mihm, J., Loch, C. H., and Huchzermeier, A. 2003. Problem-solving oscillations in complex projects. *Manag. Sci. 49*, 6, 733–750.

Mookerjee, V., Manino, M., and Gilson, R. 1995. Improving the performance stability of inductive expert systems under input noise. *Inf. Syst. Res. 6*, 328–356.

Morrice, D. J., Anderson, E. G., and Bharadwaj, S. 2004. A simulation study to assess the efficacy of linear control theory models for the coordination of a two-stage customized service supply chain. In *Proceedings of the Winter Simulation Conference*. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, Eds.

Oppenheim, A.,Willsky, A. S., and Young, I. T. 1983. *Signals and Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Parker, G. G. and Anderson, E. G. 2003. From buyer to integrator: The transformation of the supply chain manager in the vertically disintegrating firm. *Prod. Oper. Manag. 11,* 1, 75–91.

Pmi-Pmbok. 2008. *A Guide to the Project Management Body of Knowledge* 4^{th} Ed. Project Management Institute Communications.

Pipino, L. L., Lee, Y. W., and Wang, R. Y. 2002. Data quality assessment. *Comm. ACM*, *45,* 4, 211–218.

Redman, T. C. 1996. *Data Quality for the Information Age.* Artech House, Boston, MA.

Sahoo, N., Krishnan, R., Duncan, G., and Callan, J. 2011. The halo effect in multi-component ratings and its implications for recommender systems: The case of yahoo! movies. *Inf. Syst. Res. 23,* 1, 231–246.

Sethi , S. P. and Thompson, G. L. 2000. *Optimal Control Theory: Applications to Management Science and Economics.* Kluwer Academic.

Shankaranarayanan, G., Ziad, M., and Wang, R. Y. 2003. Managing data quality in dynamic decision environment: An information product approach. *J. Datab. Manag. 14,* 14–32.

Smith, R. and Eppinger, S. D. 1997. Identifying controlling features of engineering design iteration. *Manag. Sci. 43*, 3, 276–293.

Stengel, R. F. 1994. *Optimal Control and Estimation*. Dover, New York.

Sterman, J. D. 1989. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Manag. Sci. 35,* 321–339.

Tirole, J. 1993. *The Theory of Industrial Organization*. MIT Press, Cambridge, MA.

Trigg, D. W. and Leach, A. G. 1967. Exponential smoothing with an adaptive response rate. *Oper. Res. Quart. 18*, 1, 53–59.

Upton, D. M. and Staats, B. R. 2006. Lean at wipro technologies. Case study 9-607-032, Harvard Business School.

Ulrich, K. T. and Eppinger, S. D. 2000. *Product Design and Development.* McGraw-Hill, New York.

Wang, R. Y. 1998. A product perspective on total quality management. *Comm. ACM 41,* 2, 58–65.

Wang, R. Y. and Strong, D. M. 1996. Beyond accuracy: What data quality means to data consumers. *J. Manag. Inf. Syst. 12,* 4, 5–34.

Yassine, A., Joglekar, N. R., Braha, D., Eppinger, S. D., and Whitney, D. 2003. Information hiding in product development: The design churn effect. *Res. Engin. Des. 14,* 145–161.

Received March 2012; revised June 2012; accepted January 2013