

RESEARCH NOTES AND COMMENTARIES

DOES TECHNOLOGICAL MODULARITY SUBSTITUTE FOR CONTROL? A STUDY OF ALLIANCE PERFORMANCE IN SOFTWARE OUTSOURCING

AMRIT TIWANA*

College of Business, Iowa State University, Ames, Iowa, U.S.A.

Although control is presumed to be necessary to curb opportunism, its implementation in alliances can be costly and challenging. Paradoxically, some contemporary firms have counterintuitively developed successful alliances without extensive formal control. A widespread but untested assertion that might help reconcile this contradiction is that technological modularity reduces the need for alliance control. The objective of this study is to develop and test this assertion. Using data from 120 software outsourcing alliances, we show that, process control, outcome control, and modularity independently enhance alliance performance. However modularity and control are imperfect substitutes: modularity lowers the influence of process control but not of outcome control on alliance performance. Our theoretical development and empirical testing of the interactions of alliance control with modularity has significant implications for strategy theory and practice, which are also discussed. Copyright © 2008 John Wiley & Sons, Ltd.

INTRODUCTION

Although control can be costly and challenging in knowledge-intensive outsourcing alliances, it is often presumed necessary to curb opportunism (Oxley and Sampson, 2004). Control refers to the mechanisms that govern the actions of the

outsourcer firm ('controller') in a manner that furthers the interests of the outsourcing firm ('controlled'). Paradoxically, and contrary to received wisdom, some contemporary firms (e.g., Amazon.com, ebay, and Google) have established alliance portfolios that are successful without much formal control (Schonfeld, 2005).¹ A less apparent

Keywords: outsourcing alliances; modularity; control; software; alliance governance; knowledge intensive alliances

*Correspondence to: Amrit Tiwana, College of Business, Iowa State University, 2340 Gerding Business Building, Ames IA 50011-1350, U.S.A. E-mail: tiwana@iastate.edu

¹ For example, Amazon.com derived 40 percent of its \$11 billion revenue in 2006 through sales generated by 500,000 alliance partners (whom it calls affiliates) whose services and applications were built and operated without Amazon's oversight. These alliance partners conceptualize, implement, and operate independent Web sites and software applications that generate sales for Amazon. In 2006, 210,000 independent Web sites using Amazon's modular 'application programming interface' provided complementary services and generated additional sales

commonality in these counterintuitive examples is their technological modularity, which refers to the intentional decoupling of interoperating sub-systems of a larger system.

A widespread but untested assertion that may reconcile this contradiction is that technological modularity ‘substitutes’ for alliance control. Two variables are substitutes if more of one makes more of the other less valuable (Poppo and Zenger, 2002; Siggelkow, 2002). Substitutes are characterized by a negative interaction effect and represent the conceptual opposite of complements, which are characterized by a positive interaction effect. The idea that modularity and control are substitutes surfaces most notably in Sanchez’s (1995) assertion that there is little need for overt managerial supervision over development processes involving modular products. Sanchez argues that the ‘embedded coordination’ mechanisms provided by modular designs can ‘achieve coordination by means other than the overt exercise of managerial authority’ (Sanchez, 1995: 146). Although technological modularity is widely recognized as an important antecedent to product development performance within firms (Worren, Moore, and Cardona, 2002), the alliances literature has rarely explored its linkages with formal control (Hoetker, Swaminathan, and Mitchell, 2007).

In isolation from the modularity literature, control is recognized as an important antecedent to alliance performance (Mjoen and Tallman, 1997; Tiwana and Keil, 2007). Besides being costly to implement, effective interfirm control also requires ‘peripheral knowledge’ in the domains of alliance partners (Tiwana and Keil, 2007), which conflicts with the premise of specialization espoused by the core competence perspective and knowledge-based view of the firm. If modularity indeed lowers the need for formal control, it can provide a less costly, equifinal alliance governance mechanism. The question of whether modularity lowers the need for alliance control—the focus of this study—therefore has considerable theoretical implications.

Using data from 120 software outsourcing alliances, we show that technological modularity and alliance control are *imperfect substitutes*: Modularity lowers the need for process-based

but not for outcomes-based formal control, all of which independently enhance alliance performance. The theoretical elaboration and empirical testing of the effects and interactions of modularity and alliance control on alliance performance forms our central contribution.

Subsequent sections of this article develop the hypotheses put forward, describe the methodology used; outline our analyses and results; and discuss the contributions and implications of the study.

THEORY AND HYPOTHESES

Alliance control

Control refers to the mechanisms that govern actions by the controllee (the agent) in a manner that furthers the interests of the controller (the principal) (Ouchi and Maguire, 1975). The controller in software outsourcing alliances represents the firm that outsources the technology development project, and controllee represents the firm to which the project has been outsourced. Formal control is implemented by the controller using two types of mechanisms: (a) *outcome control*, which refers to the prespecification by the controller of desired controllee outputs and the associated evaluation criteria, and (b) *process control*, which refers to prescribing the methods and procedures that the controllee should follow (Kirsch *et al.*, 2002; Tiwana and Keil, 2007). Outcome control uses a market-like system to encourage the controller to achieve desired outcomes, while process control attempts to coordinate controllee actions in a manner that the controller believes will increase the likelihood of achieving them. The controllers’ incentives are primarily getting the project completed on time, within budget, and fulfilling project objectives. The controllee’s incentives include bonuses (penalties) for timely (delayed) project completion within budget and quality expectations, and increasing revenues in the short term. Such bonuses and penalties are often used both in projects with fixed-price as well as time-and-materials (variable price) contracts. Longer term incentives include developing a favorable reputation in the market as well as attracting repeat business for future projects. Outcome controls thus prespecify *what* the controllee should accomplish and process controls describe *how* the controllee should achieve those outcomes. Formal

for Amazon. For details, see <http://aws.amazon.com>. Representative examples of such partners include charlierose.com, everydeal.com, slickfillers.net, and 43folders.com.

controls therefore align the controllee's incentives with the controller's and facilitate monitoring and direction of controllee activities. Therefore, we expect both types of formal control mechanisms to enhance alliance performance.

Hypothesis 1a: Outcome control is positively associated with alliance performance.

Hypothesis 1b: Process control is positively associated with alliance performance.

Technological modularity

The notion of modularity is grounded in Simon's premise that any complex system is composed of distinct interacting subsystems that are always to some extent interdependent and independent (Karim, 2006). A complex system is said to exhibit modularity if its constituent subsystems can be designed independently but will work together to support the whole (Parnas, 1972). So modularity refers to the degree of intentional decoupling among such constituent subsystems. Modularity is achieved by: (a) codifying the interface specifications for how a subsystem connects with a larger system, and (b) minimizing their interdependencies (Sanchez and Mahoney, 1996; Sosa, Eppinger, and Rowles, 2004). The focus in interfirm alliances is on modularity across the subsystems spanning the interfirm boundary. In software outsourcing alliances, such modularity is across the boundary between the outsourced subsystem and the portfolio of software applications with which it must eventually interoperate (i.e., communicate, exchange data, or derive functionality).² We label the former as the *outsourced subsystem* and the latter as the *base system* for the remainder of this discussion. We therefore define technological modularity as *the degree of decoupling between the outsourced subsystem and the base system*, which is consistent with a general systems view of modularity as the coupling between a subsystem and the context in which it will be used (Schilling, 2000). The independence engendered by modularity provides the controllee (usually a specialist in the outsourced activity) greater autonomy to utilize

its own tacit knowledge and organizational routines to accomplish outsourced development tasks, enhancing alliance performance. This leads to our next hypothesis.

Hypothesis 2: Technological modularity is positively associated with alliance performance.

Interaction between technological modularity and control

Modularity and process control

Process control requires monitoring the controllee for compliance with prescribed procedures during the development process. In alliances, the weaker formal authority over the controllee, as compared to internal development, difficulty of monitoring, and absence of natural interfirm informational conduits make this challenging and costlier (Choudhury and Sabherwal, 2003; Tiwana and Keil, 2007). Modularity provides what Sanchez (1995: 146) characterizes as an alternative 'embedded coordination' mechanism that lowers the need for continual overt supervision. By embedded coordination, he means that the development of the outsourced subsystem can be coordinated by simply requiring conformance with the codified subsystem-base system interface specifications. Modularization thus reduces dependencies at system-subsystem interface, which is the primary source of interfirm interdependence in technology development projects (Sosa *et al.*, 2004). Changes in the internal design of the outsourced subsystem are less likely to disrupt or require changes in the base system, largely because of the reduction of the outsourced subsystem's dependencies with the base system. Overall, such lowering of interfirm interdependencies reduces the need for overt coordination and direction by the controller (Hoetker *et al.*, 2007; Sanchez, 1995; Sosa *et al.*, 2004). Thus, reduction in interfirm task interdependence allows a modular subsystem to be developed in a more autonomous fashion by the controllee without close monitoring and direction by the controller.

In contrast, design decisions in a less modular subsystem project can impact the base system, often requiring intense interfirm supervision, direction, iteration, and coordination to ensure interoperability (Sanchez and Mahoney, 1996; Smith and

² For example, a new Web-based university course registration application might interact and exchange data with a variety of other applications, such as student records, fee payment processing, email databases, online phonebooks, and classroom scheduling.

Eppinger, 1997). The reduced interfirm interdependence gained through modularity between an outsourced subsystem and the base system therefore lowers the need for process control. Two recent studies in the computer (Hoetker, 2006) and auto parts (Hoetker *et al.*, 2007) industries support this perspective by showing that suppliers of highly modular auto and computer subsystems benefited more from greater autonomy vis-à-vis suppliers of less modular parts. Modularity also lowers the threat of opportunism (Hoetker, 2006), which is the other intent of formal control. Technological modularity therefore serves the same purpose as process control in interfirm alliances, and the simultaneous use of both is not only redundant but can also trigger performance-decreasing consequences. Thus, modularity lowers the effect of process control on alliance performance.

Modularity and outcome control

Outcome control is based on rewarding or penalizing the controllee based on how well the outcomes of its development process meet those that the controller established at the outset (e.g., delivery schedule, project milestones, cost, and project objectives). Outcome controls therefore establish accountability for the controllee's outputs using output evaluation as a governance instrument. Although an increase in technological modularity lowers the need for overtly controlling the outsourced development process, it does not decrease the need for its outcomes to meet the controller's specifications. The use of outcome controls will therefore enhance alliance performance independent of the level of modularity of the outsourced subsystem. Likewise, technological modularity, by decreasing interdependence across the controller-controllee interfirm boundary, will also enhance alliance performance independent of the extent of outcome controls used to govern alliance activities. Modularity thus only reduces the need for control over the development process but does not ensure that controller objectives (e.g., cost, schedule) are met. Therefore, even though technological modularity and outcome control enhance alliance performance, we expect their effects to be distinct and mutually independent. The foregoing arguments therefore lead to our final hypothesis.

Hypothesis 3: Technological modularity and control are imperfect substitutes, i.e., modularity

lowers the effect of process control but not of outcome control on alliance performance.

METHODOLOGY

Data collection

Data were collected in two phases through a field study of 120 alliances in which a U.S. firm outsourced custom development of a software application to an Indian software firm. Such alliances are an increasingly common interfirm arrangement for accessing specialized technical software design and development capabilities (Ethiraj *et al.*, 2005). This research context exhibits the contextual characteristics in which Sanchez and Mahoney (1996) ground their modularity arguments: (a) software applications can have large variance in modularity, (b) such applications represent a project-based development process, and (c) the fact that the controller and controllee are geographically dispersed make control both more important and challenging. Our sampling frame was 627 Indian software service firms who were members of India's largest software industry consortium, National Association of Software and Service Companies (NASSCOM). We received responses from 120 lead project managers from 120 independent firms for a 19.1 percent response rate. We approached the chief executives of all 719 NASSCOM member firms that listed software services outsourcing as their specialty and asked them to identify the lead project manager for a project outsourced by a U.S. company, of which 627 were tentatively willing to participate. T-tests comparing the early (first 40) and late (last 40) respondents provided assurance against nonresponse bias.

Construct measures and control variables

Existing seven-point Likert scales were used for all key constructs in the model, except the new technological modularity scale (see Appendix for measures, including control variables). The unit of analysis was the outsourced project. The controller and controllee referred to project participants from the outsourcing and outsourcee firms respectively. The iteratively refined *technological modularity* scale tapped into the extent to which the relationships between the outsourced subsystem and the base system were characterized by:

(1) high interoperability, (2) stable, well-defined interfaces, (3) well-understood interdependencies, and (4) minimal unnecessary interdependencies, consistent with our theoretical definition. We started with Nambisan's (2002) *modularity decision* scale developed in the software context and Worren *et al.*'s (2002) *modular product* scales. We refined the item pool based on Sanchez and Mahoney's (1996) conceptualization of product modularity, which we iteratively refined through feedback from seven software industry professionals and five academics to ensure meaningfulness and unequivocality in our study's context. Kirsch *et al.*'s (2002) scales for *outcome control* and *process control* (labeled 'behavior control' in their study) were used with minor adaptations. *Alliance performance* used four items adapted from Faraj and Sproull's (2000) scale, consistent with software alliance performance criteria (e.g., Ethiraj *et al.*, 2005; Tiwana and Keil, 2007). The constructs exhibited sufficient discriminant validity (verified using factor analysis) and reliability ($\alpha \geq 0.7$) (Nunally, 1978). The construct correlations, descriptives, and reliability coefficients are summarized in Table 1.

Control variables

Ten control variables spanning *alliance characteristics* (interfirm collaborative history, relational governance, and interfirm ties), *project characteristics* (alliance scope, alliance goal tacitness, contract structure, project duration, and team size), *firm characteristics* (such as national origin), and *industry characteristics* were used to account for rival explanations of alliance performance. Since interfirm alliance history can influence alliance functioning (Oxley and Sampson, 2004), we controlled for *controller-controllee collaborative history* using a dummy variable. Controllers can complement formal control with *relational governance* mechanisms (Poppo and Zenger, 2002) or rely on strong interfirm *controller-controllee ties*; which were included as controls.

Among the project characteristics, we controlled for *alliance scope*, recognizing that projects with scope larger than what is typical for the controllee firm are more challenging to manage (Ethiraj *et al.*, 2005). Greater *alliance goal tacitness* makes it harder to communicate the project's goals to the controllee, potentially lowering alliance performance. We also included a dummy for

time-and-materials contracts, which do not penalize the controllee for controller-initiated modifications like fixed-price contracts do (Ethiraj *et al.*, 2005). We also controlled for *project duration*. Finally, we controlled for *team size* (Ethiraj *et al.*, 2005), premising that a larger team is representative of a larger pool of dedicated controllee resources. We used team size instead of firm size (for which 2006 public data were unavailable; it was non-significant with 2004 firm size data ($\beta = 0.02$, T-value = 0.85)).

We also controlled for controllee *national origin* (indigenous Indian firms coded as 1; subsidiaries of non-Indian firms as 0). This accounts for variance due to a more expansive global pool of resources and skills potentially accessible to a nonindigenous multinational firm. Although additional firm-level controls might have been appropriate, current firm-level data were not available in public records for the many privately held firms in the study and were not captured in the survey to economize on questionnaire length. To account for differences in software complexity across industries, the industry dummy variables used by Ethiraj *et al.* (2005) were included (*financial services, manufacturing, retail; other* was omitted).

Demographics

The controller firms in the sample represented a diverse variety of industries (financial services, 18%; manufacturing, 15%; retail, 5%), and a majority (56.1%) of them had prior project experience with the controllee. Since a large percentage (62%) of the dyads were in the *other* category, we retested the model using *other* as an industry dummy in the analysis; the results remained consistent with Table 2. Indigenous Indian firms comprised 73.1 percent of the controlees in the study. The primary informants, the project managers, were highly experienced (average IT experience 12.3 years, sd 6.7). The majority (74%) of alliances used fixed price contracts. The average project duration was 12.8 months (sd 11.1) and team size was 25.8 individuals (sd 49.5).

ANALYSES AND RESULTS

The analyses used an ordinary least squares hierarchical regression model in which the control variables were introduced (Step 1), followed by

Table 1. Descriptive statistics and construct correlation matrix

Construct	Mean	S.D.	α	Correlations															
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1. Interfirm collaborative history ⁺	—	—	—																
2. Relational governance	5.3	1.2	0.85	0.18															
3. Controller-controllee interfirm ties	6.2	0.7	0.85	-0.02	0.40**														
4. Project alliance scope	4.2	1.3	0.90	0.03	0.16	0.10													
5. Alliance goal tacitness	2.8	1.2	0.84	-0.13	-0.24*	-0.20*	-0.02												
6. Time-and-materials contract ⁺	—	—	—	0.18	0.06	0.00	0.18	0.10											
7. Project duration	12.8	11.1	—	-0.04	0.22*	0.01	0.30**	-0.02	0.19										
8. Project team size	25.8	49.5	—	-0.09	0.11	0.05	0.17	-0.07	-0.06	0.34**									
9. Controllee national origin ⁺	—	—	—	-0.16	-0.11	-0.02	-0.08	-0.01	-0.20*	0.04	0.04								
10. Controller industry - Financial services ⁺	—	—	—	0.07	0.04	0.10	-0.09	-0.02	0.00	0.15	0.17	0.13							
11. Controller industry - Manufacturing ⁺	—	—	—	-0.16	-0.08	-0.08	0.12	0.20*	-0.03	-0.15	-0.13	0.08	-0.13						
12. Controller industry - Retail ⁺	—	—	—	0.02	0.01	0.09	-0.22*	0.04	-0.15	-0.13	-0.08	0.13	-0.10	-0.09					
13. Outcome control	6.1	0.7	0.70	-0.02	0.15	0.13	-0.05	0.01	-0.06	-0.04	0.04	-0.04	-0.07	0.20*	-0.01				
14. Process control	5.7	1.1	0.87	-0.09	0.30**	0.32**	0.06	-0.05	0.13	0.02	-0.03	-0.01	-0.03	0.19*	0.00	0.10			
15. Modularity	5.3	1.0	0.72	0.05	0.24*	0.27**	-0.02	-0.16	0.07	0.00	-0.05	0.00	0.02	-0.02	-0.19	0.18	0.43**		
16. Alliance performance	6.2	0.7	0.89	-0.12	0.16	0.43**	0.03	-0.28**	-0.09	-0.03	-0.12	0.17	-0.03	0.02	0.09	0.15	0.42**	0.44**	

** p < 0.01, * p < 0.05 two-tailed test; + dummy variable; N = 120 alliances

Table 2. Effects of technological modularity and control on alliance performance

	Step 1		Step 2		Step 3.1		Step 3.2		Step 3.3	
	Control variables		Main effects		Interaction terms		Interaction terms		Interaction terms	
	β (T-value)	S.E.	β (T-value)	S.E.	β (T-value)	S.E.	β (T-value)	S.E.	β (T-value)	S.E.
(Constant)	(4.35)	0.78	(2.03)	0.70	(2.11)	0.73	(2.27)	0.68	(2.31)	0.87
Interfirm collaborative history	-0.12(-1.09)	0.16	-0.06(-0.70)	0.12	-0.06(-0.74)	0.12	-0.04(-0.50)	0.12	-0.04(-0.51)	0.12
Relational governance	0.05(0.45)	0.07	-0.08(-0.90)	0.06	-0.08(-0.89)	0.06	-0.08(-0.97)	0.05	-0.09(-1.05)	0.05
Controller-controlled interfirm ties	0.41** (3.61)	0.12	0.06(0.61)	0.11	0.06(0.59)	0.11	0.05(0.50)	0.10	0.03(0.29)	0.10
Project alliance scope	0.08(0.68)	0.06	0.22** (2.52)	0.05	0.21** (2.39)	0.05	0.21** (2.54)	0.05	0.23** (2.68)	0.05
Alliance goal tacitness	-0.17(-1.53)	0.07	-0.10(-1.21)	0.05	-0.10(-1.20)	0.05	-0.07(-0.93)	0.05	-0.07(-0.81)	0.05
Time-and-materials contract	-0.03(-0.29)	0.18	-0.12(-1.34)	0.14	-0.11(-1.27)	0.14	-0.12(-1.43)	0.14	-0.14(-1.63)	0.14
Project duration	0.00(-0.01)	0.01	0.03(0.29)	0.01	0.03(0.31)	0.01	0.00(0.00)	0.01	0.00(-0.02)	0.01
Project team size	-0.19(-1.56)	0.00	-0.21* (-2.23)	0.00	-0.20* (-2.21)	0.00	-0.19* (-2.25)	0.00	-0.19* (-2.26)	0.00
Controllee national origin	0.18(1.61)	0.18	0.13(1.58)	0.14	0.13(1.57)	0.14	0.10(1.28)	0.13	0.11(1.38)	0.13
Controller industry - Financial services	-0.06(-0.52)	0.20	0.02(0.22)	0.15	0.01(0.15)	0.15	-0.01(-0.12)	0.14	-0.01(-0.13)	0.14
Controller industry - Manufacturing	0.04(0.37)	0.22	0.00(0.00)	0.17	-0.01(-0.08)	0.18	-0.03(-0.34)	0.17	-0.03(-0.42)	0.17
Controller industry - Retail	0.03(0.25)	0.37	0.17* (1.83)	0.31	0.16* (1.77)	0.31	0.13(1.48)	0.29	0.12(1.41)	0.29
Outcome control			0.28** (2.84)	0.11	0.27** (2.70)	0.11	0.27** (2.90)	0.10	0.20(1.55)	0.14
Process control			0.32*** (3.22)	0.07	0.33*** (3.26)	0.07	0.33*** (3.56)	0.06	0.32*** (3.36)	0.07
Modularity			0.30** (3.02)	0.07	0.29** (2.86)	0.07	0.30*** (3.26)	0.07	0.35*** (3.29)	0.08
Outcome control x Process control				0.05	-0.05(-0.63)	0.05	0.09(1.05)	0.06	0.16(1.36)	0.08
Modularity x Process control							-0.28*** (-3.21)	0.06	-0.22* (-1.86)	0.08
Modularity x Outcome control									-0.17(-0.87)	0.13
R ² (Model F)	33.12%** (2.72)		63.10%*** (7.18)		63.33%*** (6.69)		68.63%*** (7.85)		69.02%*** (7.43)	
R ² Adj	20.95%		54.31%		53.87%		59.88%		59.73%	
ΔR^2 (F-change)	—		29.98%*** (17.06)		0.23%(0.39)		5.30%*** (10.30)		0.39%(0.76)	

***p < 0.001, ** p < 0.01, * p < 0.05, one-tailed tests; significant in bold; N = 120 project alliances

the main effects (Step 2), and finally the residual-centered interaction terms entered sequentially (Step 3.1 through 3.3). We used Lance's (1988) residual centering procedure for computing the interaction terms to correct the problem of partial coefficient distortion faced in the simultaneous analysis of main effects and interaction terms due to their correlation (Jaccard and Turrisi, 2003: 27). Using a two-stage procedure, we first regressed each product term (e.g., process control \times modularity) on its components (e.g., process control and modularity). We then used the resulting residual instead of the interaction term, which reduced multicollinearity between the main effects and the interaction terms. The resulting regression coefficient can be interpreted directly as the effect of the interaction term on the dependent variable. The results are summarized in Table 2.

The control variables accounted for 20.95 percent of the variance. Notably, interfirm history was nonsignificant in spite of its prominence in the alliances literature. This is unsurprising given Hoetker's (2006) recent finding that history is nonsignificant in alliances involving modular technologies (as our sample's modularity mean of 5.32 indicates) but significant in non-modular technologies. Of the main effects, outcome control ($\beta = 0.28$, T-value 2.84, $p < 0.01$), process control ($\beta = 0.32$, T-value 3.22, $p < 0.001$), and modularity ($\beta = 0.30$, T-value 3.02, $p < 0.01$) were significant and positive. This suggests that independently both forms of formal control and modularity enhance alliance performance, supporting Hypotheses 1a, 1b, and 2. The larger coefficient of process control suggests that it is a more important predictor of alliance performance than outcome control. Next, the interaction term between process control and outcome control was added in Step 3.1. Although this interaction is not hypothesized, this term must be added to ensure that the interaction model is not underspecified (see Jaccard and Turrisi, 2003). The absence of an interaction between outcome and process control in Step 3.1 is consistent with prior evidence of their independence (Ouchi and Maguire, 1975).

Hypothesis 3 regarding imperfect complementarities between modularity and control proposed

that modularity lowers the effect of process control but not of outcome control on alliance performance. This requires a: (a) negative and significant interaction between modularity and process control and (b) a nonsignificant interaction between modularity and outcome control. The interaction between modularity and process control is significant and negative ($\beta = -0.28$, T-value -3.21 , $p < 0.001$; Step 3.2 in Table 2) and the interaction between modularity and outcome control is nonsignificant ($\beta = -0.17$, T-value -0.87 , ns; Step 3.3). Thus, Hypothesis 3 is supported. The main effects contributed 33.36 percent and the interaction terms 5.42 percent to the explained variance in alliance performance, with a significant ΔR^2 at each key hypothesized step.

Tests for common methods bias

We conducted four types of statistical analyses to assess the threat of common methods bias. First, we conducted *Harman's one-factor test*, where the emergence of a single factor that accounts for a large proportion of the variance in factor analyses suggests a common methods bias (Podsakoff *et al.*, 2003). However, no such single factor emerged, and the first factor accounted for 13.1 percent of the 75.7 percent explained variance. Second, the *Lindell-Whitney marker variable test* uses a theoretically unrelated *marker* variable to adjust the correlations among the principal constructs (Lindell and Whitney, 2001). Any high correlation among any of the study's principal constructs and the marker variable would indicate common methods bias (Malhotra, Kim, and Patil, 2006). For robustness, we repeated the test with three different marker variables. We separately repeated the test with three dummy marker variables theoretically unrelated to our principal constructs: (1) the existence of controllee firm operations in South America, (2) the count of vertical industry segments in which the controllee firm operated, and (3) whether the project's software platform was Microsoft Windows. The average correlation between the study's principal constructs for South American operations ($r = .003$, $T = .246$), vertical segment count ($r = .003$, $T = .025$), and Windows ($r = -.015$, $T = -.084$) was low and nonsignificant, providing no evidence of common methods bias. Third, the pairwise correlation

matrix in Table 1 did not indicate any exceptionally correlated variables. The highest correlation among the principal constructs is 0.44 (i.e., below Bagozzi, Yi, and Phillips's [1991] 0.8 threshold). Finally, we cross-validated controller-controllee performance evaluations for the subsample for which matched pair data were collected. Since our attempt to collect matched pairs for alliance performance from the primary project liaison in the client firm for each project yielded only a 29.7 percent response rate (30 usable responses from the 101 projects for which the Indian alliance partner did not decline cooperation), the matched pair sample size is too small to directly use in the model, but sufficient to assess controller-controllee rater agreement. The correlation between the two respondents was positive and statistically significant ($r = .637$, $T\text{-value} = 4.14$, $p < 0.001$), suggesting strong interrater agreement. No outsourcer-side respondent was responsible for more than one project in the study. Collectively, these four tests suggest that common methods bias is not a serious threat. We therefore used data from the 120 project managers for the analyses.

Limitations

The results should be interpreted cognizant of four caveats. First, some firm characteristics such as firm size, revenue, age, and size of internal IT departments that we failed to control for are plausible alternative explanations of alliance performance; this represents a serious limitation of this study. Second, although organizational modularity is a complementary facet of technological modularity, we excluded it from our model because alliance formation itself represents organizational modularization (Schilling, 2000). Third, our measures of formal control represent received control rather than attempted control. Finally, the projects in the study were highly asset-specific and likely not mass market (e.g., automobiles and appliances) since they are customized to idiosyncratic controller needs. Caution is therefore warranted in generalizing our results.

DISCUSSION AND IMPLICATIONS

This study sought to test the widespread assertion that modularity represents an alternative to control as an alliance governance device, and that greater

use of one lessens the effect of the other. If modularity indeed substitutes for control, it can provide a less costly, autonomy-preserving alliance governance device. While such substitution is widely asserted in the modularity literature, it has neither been theoretically developed in the alliance context nor empirically tested.

We examined both forms of formal control i.e., outcome control and process control. Analyses of data from 120 software outsourcing alliances show that process control, outcome control, and modularity independently enhance alliance performance. However, their interactions show that modularity and control are imperfect substitutes: modularity lowers the effect of process control but not of outcome control on alliance performance. Modularity and outcome control instead have mutually independent effects on alliance performance. The paper's three main contributions stem from these findings.

The finding that modularity and both forms of formal control independently enhance alliance performance confirms in an alliance context what the broader modularity and organizational control literatures have found in non-alliance contexts. The distinctive contributions of this study however are with regard to the interactions between modularity and control. The finding that modularity lowers the effect of process control on alliance performance suggests that modularity can provide an alternative to process control as an alliance governance mechanism. This finding also contributes a theoretical explanation and evidence for Hoetker *et al.*'s (2007) speculation that modularity can lower alliance governance costs. However, our results also show that modularity does not decrease the need for outcome control.

These results have two important theoretical implications. First, knowledge-intensive alliances often create a legitimate risk of appropriation of technology by alliance partners (Oxley and Sampson, 2004). Modularity can mitigate this hazard by lowering the need for intensive interfirm interactions during the development process and increasing causal ambiguity about the base system by restricting controllees' exposure largely to the outsourced subsystem. These findings can cautiously be generalized to other knowledge-intensive activities that can be modularized and are increasingly

outsourced (e.g., prescription fulfillment, call centers, claims processing, and tax preparation). Second, our distinction of process from outcome control helps refine Sanchez and Mahoney's (1996) assertion that modular organizational structures are more appropriate for developing modular products, which recent empirical studies have failed to support (e.g., Brusoni and Prencipe, 2006; Hoetker, 2006). Our findings imply that modular product development demands *semimodular* organizational arrangements where developers can be granted autonomy over the development process but not over outcomes.

These findings lead to four questions for future research. First, firms need advanced architectural knowledge to accomplish modularization (Sanchez and Mahoney, 1996). How can they then manage the tension between maintaining a broader knowledge base yet specialize narrowly in their domain of core competence? Second, does excessive modularity, by isolating the outsourcing firm from developments and working knowledge of outsourced modular components, create blind spots that eventually handicap its ability to innovate? Third, does modularity indeed reduce interfirm knowledge spillovers as asserted here and in the modularity literature? Finally, technological modularity provides the real option (defined as the right without obligation) to shift the locus of complementary innovation to a broad array of specialized partners in the open market. The relationship between technological modularity and embedded real option value therefore represents a promising opportunity for theory development.

In conclusion, our overarching implication is that while both technological modularity and control independently enhance alliance performance, increasing modularity can provide a viable alternative to process control. More broadly, these findings contribute to a burgeoning literature that emphasizes the often-underappreciated interactions among managers' strategic choices.

ACKNOWLEDGEMENTS

The author is grateful for developmental feedback from Editor Will Mitchell and the anonymous reviewers.

REFERENCES

- Bagozzi RP, Yi Y, Phillips LW. 1991. Assessing construct validity in organizational research. *Administrative Science Quarterly* **36**(3): 421–458.
- Barki H, Rivard S, Talbot J. 2001. An integrative contingency model of software project risk management. *Journal of Management Information Systems* **17**(4): 37–69.
- Brusoni S, Prencipe A. 2006. Making design rules: a multidomain perspective. *Organization Science* **17**(2): 179–189.
- Choudhury V, Sabherwal R. 2003. Portfolios of control in outsourced software development projects. *Information Systems Research* **14**(3): 291–314.
- Ethiraj S, Kale P, Krishnan M, Singh J. 2005. Where do capabilities come from and how do they matter? A study in the software services industry. *Strategic Management Journal* **26**(1): 25–45.
- Faraj S, Sproull L. 2000. Coordinating expertise in software development teams. *Management Science* **46**(12): 1554–1568.
- Hansen M. 2002. Knowledge networks: explaining effective knowledge sharing in multiunit companies. *Organization Science* **13**(3): 232–248.
- Hoetker G. 2006. Do modular products lead to modular organizations? *Strategic Management Journal* **27**(6): 501–518.
- Hoetker G, Swaminathan A, Mitchell W. 2007. Modularity and the impact of buyer-supplier relationships on the survival of suppliers. *Management Science* **53**(2): 171–191.
- Jaccard J, Turrisi R. 2003. *Interaction Effects in Multiple Regression* (2 edn). Sage: Thousand Oaks, CA.
- Karim S. 2006. Modularity in organizational structure: the reconfiguration of internally developed and acquired business units. *Strategic Management Journal* **27**(9): 799–823.
- Kirsch L, Sambamurthy V, Ko D, Purvis R. 2002. Controlling information systems development projects: the view from the client. *Management Science* **48**(4): 484–498.
- Lance C. 1988. Residual centering, exploratory and confirmatory moderator analysis, and decomposition of effects in path models containing interactions. *Applied Psychological Measurement* **12**(2): 163–175.
- Lindell M, Whitney D. 2001. Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology* **86**(1): 114–121.
- Malhotra N, Kim S, Patil A. 2006. Common method variance in IS research: a comparison of alternative approaches and a reanalysis of past research. *Management Science* **52**(12): 1865–1883.
- Mjoen H, Tallman S. 1997. Control and performance in international joint ventures. *Organization Science* **8**(3): 257–274.
- Nambisan S. 2002. Complementary product integration by high technology new ventures: the role of initial technology strategy. *Management Science* **48**(3): 382–398.
- Nunnally J. 1978. *Psychometric Theory*. McGraw-Hill: New York.

- Ouchi W, Maguire M. 1975. Organizational control: two functions. *Administrative Science Quarterly* **20**: 559–569.
- Oxley JE, Sampson RC. 2004. The scope and governance of international R&D alliances. *Strategic Management Journal*, August–September Special Issue **25**: 723–749.
- Parnas D. 1972. On the criteria to be used in decomposing systems into modules. *Communications of the ACM* **15**(9): 1053–1058.
- Podsakoff P, MacKenzie S, Lee J, Podsakoff N. 2003. Common method biases in behavioral research: a critical review of the literature and recommended remedies. *Journal of Applied Psychology* **88**(5): 879–903.
- Poppo L, Zenger T. 2002. Do formal contracts and relational governance function as substitutes or complements? *Strategic Management Journal* **23**(8): 707–725.
- Sanchez R. 1995. Strategic flexibility in product competition. *Strategic Management Journal*, Summer Special Issue **16**: 135–159.
- Sanchez R, Mahoney JT. 1996. Modularity, flexibility, and knowledge management in product organization and design. *Strategic Management Journal*, Winter Special Issue **17**: 63–76.
- Schilling M. 2000. Toward a general modular systems theory and its application to interfirm product modularity. *Academy of Management Review* **25**(2): 312–334.
- Schonfeld E. 2005. The Great Giveaway. *Business 2.0* April: 81–86.
- Siggelkow N. 2002. Misperceiving interactions among complements and substitutes: organizational consequences. *Management Science* **48**(7): 900–916.
- Smith R, Eppinger S. 1997. Identifying controlling features of engineering design iteration. *Management Science* **43**(3): 276–293.
- Sosa M, Eppinger S, Rowles C. 2004. The misalignment of product architecture and organizational structure in complex product development. *Management Science* **50**(12): 1674–1689.
- Tiwana A, Keil M. 2007. Does peripheral knowledge complement control? An empirical test in technology outsourcing alliances. *Strategic Management Journal* **28**(6): 623–634.
- Worren N, Moore K, Cardona P. 2002. Modularity, strategic flexibility, and firm performance: a study of the home appliance industry. *Strategic Management Journal* **23**(12): 1123–1140.

APPENDIX: CONSTRUCT MEASURES

Likert scale anchors were strongly disagree–strongly agree unless otherwise noted; seven-point scales.

Technological modularity (new scale): How well do the following characterize the relationships between the outsourced system and the client's internal IT applications: (1) highly interoperable, (2) stable, well-defined interfaces, (3) well-understood interdependencies, and (4) minimal unnecessary interdependencies.

Outcome control (adapted from Kirsch *et al.* (2002)): In this project, the client placed significant weight on: (1) timely project completion, (2) completion within budget, (3) meeting project requirements, and (4) accomplishing project goals.

Process control (adapted from Kirsch *et al.* (2002)): In this project, the client placed significant weight on an understandable written sequence of steps: (1) toward accomplishing project goals, (2) to ensure this system met client requirements, and (3) to ensure the success of this project.

Alliance performance: The development process was effective in successfully fulfilling this client's project: (1) needs, (2) quality expectations, (3) functional requirements, and (4) objectives.

Control variables

Interfirm collaborative history: Was this your company's first project for this client (one)? Yes No {coded 1}

Relational governance (adapted from Kirsch *et al.*'s (2002) clan control scale): Members of the client organization involved in this project: (1) attempted to be 'regular' members of the project team, (2) placed a significant weight on

understanding the project team's goals, values, and norms, and (3) actively participated in project meetings to understand the project team's goals, values, and norms.

Interfirm controller-controllee ties (adapted from Hansen (2002)): Our overall working relationship with the client in this project can be characterized by: (1) regular interactions, (2) frequent communication, and (3) close working relationships.

Project alliance scope (adapted from Barki, Rivard, and Talbot's (2001)) {anchors: Much smaller—Much larger}: How would you compare this project to other IT projects completed by your company? (1) person-months of development work, (2) project duration, and (3) dollar budget.

Alliance goal tacitness (adapted from Hansen's (2002) noncodification scale; reverse scored): The client's requirements for this project could easily be: (1) documented, (2) explained to us in writing, and (3) communicated formally (e.g., via documents, requirements, code comments, & manuals)

Contract structure: Type of contract used (one)? Fixed-price Time-and-materials {coded as 1}

Project duration: measured in months

Project team size: Approximately, how many individuals from your company were involved in this project?

Controllee national origin: Is your company a subsidiary of a foreign company (one)? Yes {coded as 1} No

Client industry (one)? Financial services Manufacturing Marketing Other