

RESEARCH NOTES AND COMMENTARIES

DOES INTERFIRM MODULARITY COMPLEMENT IGNORANCE? A FIELD STUDY OF SOFTWARE OUTSOURCING ALLIANCES

AMRIT TIWANA*

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

Knowledge-intensive outsourcing alliances present outsourcers with a tension between simultaneously sharing enough private knowledge to accomplish alliance goals and safeguarding such knowledge against misappropriation. This study explores the perspective that increasing interfirm modularity lowers the need for interfirm knowledge sharing. Put another way, modularity complements outsourcee ignorance. Analyses of data on 209 alliances between U.S. firms and software services firms in Russia, Ireland, and India provide strong support for this idea. Our theoretical elaboration and empirical testing of the complementarities between modularity and outsourcee ignorance has significant implications for strategy theory, which are also discussed. Copyright © 2008 John Wiley & Sons, Ltd.

INTRODUCTION

Knowledge-intensive outsourcing alliances often require integration of proprietary knowledge about the outsourcing firm's ('outsourcer') idiosyncratic business practices, routines, processes, and technologies in performing alliance activities (Brusoni, Prencipe, and Pavitt, 2001; Ethiraj *et al.*, 2005). This poses an uneasy tension for the outsourcer: How can it allow the 'outsourcee' to utilize its private knowledge in alliance activities yet safeguard against its misappropriation.

Keywords: modularity; outsourcing; ignorance; alliances; peripheral knowledge; knowledge-based view

*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

Prior research suggests two safeguards against knowledge misappropriation hazards: (1) using appropriate governance mechanisms (Gulati and Singh, 1998; Kale, Singh, and Perlmutter, 2000) and (2) reducing the scope of the alliance (Oxley and Sampson, 2004). However, these works do not provide guidance for effectively organizing alliance activities without reducing outsourcee ignorance by sharing valuable private knowledge. Contractual safeguards afford limited protection because contracts involving complex tacit knowledge are notoriously difficult to specify and enforce (Oxley and Sampson, 2004).

In this research note, we explore a complementary theoretical perspective: increasing interfirm modularity complements outsourcee ignorance. Put another way, increasing modularity substitutes for

interfirm knowledge sharing in alliances. This idea is implicitly suggested in the modularity and alliances literature (Oxley and Sampson, 2004; Sanchez and Mahoney, 1996), although no causal explanations or empirical tests of any theoretical explanations exist. If such complementarities indeed exist, modularity would mitigate knowledge misappropriation hazards by lowering the need for outsourcing firms to reveal valuable private knowledge to outsourcees. The idea is tested in a field study of 209 independent alliances in which U.S. firms outsourced the development of custom business software applications to software services firms in Russia, Ireland, and India.

Our noteworthy contribution is explaining how interfirm modularity complements outsourcee ignorance in knowledge-intensive alliances. Rather than altogether abandoning the potential benefits of outsourcing to a specialist firm, increasing interfirm modularity allows firms to achieve outsourcing alliance objectives while simultaneously mitigating knowledge appropriation hazards. This research note has provocative theoretical implications for the burgeoning literature on the knowledge-based view (KBV) of the firm, to which it adds a counterintuitive notion regarding the strategic value of ignorance rather than knowledge. The rest of the research note is structured as follows: the central idea is developed in the next section, followed by the methodology, analyses and results, and finally the contributions and implications of the study.

THEORY AND HYPOTHESIS

Software outsourcing alliances

A software outsourcing alliance is defined as an interfirm arrangement in which an *outsourcee* firm custom develops a business software application uniquely tailored to address a specific business problem in the *outsourcing* firm. The outsourcer-outsourcee distinction in the alliances literature is comparable to the buyer-supplier distinction in the marketing literature and the client-vendor distinction in the information systems literature, but with a few noteworthy differences. Unlike in traditional buyer-supplier alliances (e.g., automobile parts) where supplier outputs can be more readily specified and their quality more objectively inspected, the end product of such knowledge-intensive outsourcing alliances is usually intangible, difficult

to precisely prespecify, firm-specific, and equifinal (i.e., with many possible solutions for the same problem). Furthermore, unlike generic, shrink-wrapped packaged software applications traded in the open market (e.g., Microsoft Word or Powerpoint), such software applications are uniquely tailored to the outsourcing firm's idiosyncratic business needs. The development process in such alliances requires understanding and embodying in the software application the outsourcer's idiosyncratic private knowledge pertaining to the project's business problem domain (Adelson and Soloway, 1985; Ethiraj *et al.*, 2005; Robillard, 1999). Such private knowledge encompasses the outsourcer's idiosyncratic business processes, practices, routines, and trade secrets associated with the project (Matusik, 2002). Ethiraj *et al.* (2005) provide examples of how supply chain management systems and trading exchange software applications require an in-depth understanding of firm-specific business processes and routines. The ideal project outcome is a business software application that has high outsourcer specificity and interoperability with the outsourcer's *technological portfolio* (which is defined as the portfolio of hardware, software applications, business processes, databases, and electronic networks in the outsourcer firm with which the outsourced project has functional, procedural, or informational interdependencies) (Ethiraj *et al.*, 2005; Prahalad and Krishnan, 2002). A failure to embody the outsourcer's private knowledge in the development process can have devastating consequences on alliance performance, potentially rendering even a technically sound software application strategically irrelevant (Hoopes and Postrel, 1999).

This requires access to the outsourcer's private knowledge, which defines the context for the outsourced project and how the resulting solution eventually integrates with the outsourcer's technological portfolio. The outsourcee—while technically competent—might lack idiosyncratic knowledge of the project's business problem domain. We refer to the outsourcee firm's *lack of* such project-specific private knowledge as *outsourcee ignorance*. Successfully accomplishing alliance objectives might therefore require exposing such competitively valuable private knowledge to the outsourcee, which creates a legitimate risk of its misappropriation or leakage to competitors. The more interdependent the activities performed in the alliance, the greater is this risk (Oxley

and Sampson, 2004). This risk is especially pronounced in the software services industry, where firms compete on their depth of client domain and industry knowledge (Ethiraj *et al.*, 2005).

The optimality of ignorance

Paradoxically, ignorance is optimal from either firm’s perspective. Outsourcers have a strong disincentive to expose their private knowledge to the outsourcee because it can be an important potential source of competitive differentiation (Matusik, 2002). Likewise, software services firms incur a fixed, utilization-independent cost in acquiring knowledge that is idiosyncratic to a specific client (Becker and Murphy, 1992), which they are then even more likely to attempt to recoup through reuse in other projects for similar clients in the outsourcer’s industry. Outsourcees will ordinarily acquire knowledge about only a compact, generic set of common business problems since they facilitate communicating with a broad array of prospective clients. Once the cost of transferring idiosyncratic private knowledge from a client is incurred, the outsourcee has a stronger incentive to internalize and appropriate it in projects for other client firms, especially as competition among software services firms is now becoming increasingly dependent on their domain-specific capabilities (Ethiraj *et al.*, 2005).

Interfirm modularity

Following Sanchez and Mahoney (1996), we define interfirm modularity as the looseness of coupling between the outsourced project and outsourcing firm’s technological portfolio with which it has functional, procedural, or informational interdependencies. Interfirm modularity is achieved by specifying in detail *at the outset* of the development process details for *attachment* (how they will connect), *information exchange*, and *interactions* between the outsourced software application and the outsourcer’s technological portfolio (Mikkola and Gassman, 2003; Sanchez and Mahoney, 1996). Interfirm modularity is therefore higher if the outsourced project’s relationship with the outsourcer’s technological portfolio is characterized by weaker coupling, lower interdependence, comprehensiveness of their *ex ante* interface specifications, and if fewer changes are required in the outsourcer’s technological portfolio to compensate for internal

design changes in the outsourced system. Complex software systems typically have thousands of lines of interdependent computer code, therefore internal design changes in the outsourced project are inherently likely to create a cascade of required changes in other systems in the outsourcing firm (Messerschmitt and Szyperski, 2003). Modularity lowers such interdependence between the outsourced system and the outsourcer’s technological portfolio.

Modularization however is costly for the outsourcer because it is the intentional *ex ante* act of loosening the coupling between them that increases modularity. There are situations when an outsourcer might prefer not to modularize a software project, such as when the outsourcer lacks the technical expertise to effectively accomplish modularization or when an integral system is preferable due to performance or reliability considerations (e.g., software ‘embedded’ in a digital camera or automobile).

Complementarity between interfirm modularity and outsourcee ignorance

Figure 1 illustrates four possible combinations of interfirm modularity and ignorance. Note that these are simplistic representations because both constructs are continua rather than discrete-level constructs.

The baseline scenario (cell A) is low modularity and low ignorance. This represents the classic outsourcing model involving integral products where the outsourcee has knowledge about the project’s business problem domain. This configuration does not incur the cost of project modularization but is characterized by high levels of integrative activities across the interfirm boundary. Although this

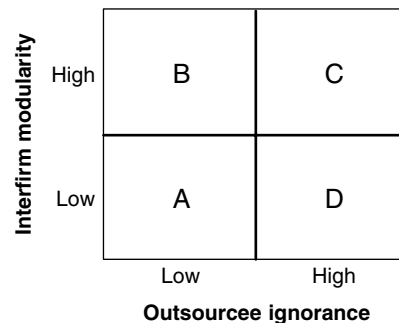


Figure 1. Four possible combinations of interfirm modularity and outsourcee ignorance

configuration has historically sufficed for manufacturing alliances and buyer-supplier relationships in spite of the cost of integrative activities, it imposes additional knowledge-related costs on both the outsourcer and outsourcee in knowledge-intensive alliances. This is because the outsourcee incurs the cost of acquiring idiosyncratic client-specific knowledge, which has uncertain future value unless it is amortized across future projects. Similarly, the outsourcer incurs the costs and bears the risks associated with sharing private knowledge with the outsourcee.

Changing one dimension of the configuration in cell A yields cells B and D. The high ignorance-low modularity configuration (cell D) is likely to create an alliance performance penalty since the high interfirm dependencies that are also poorly understood by the outsourcee require extensive interfirm integration effort (Sosa, Eppinger, and Rowles, 2004). The high modularity-low ignorance configuration (cell B) is more likely to exact a cost penalty rather than performance penalty since it incurs both modularization costs and interfirm knowledge sharing costs. Further, if outsourcee ignorance is reduced by knowledge sharing, this configuration also exposes the outsourcer to a knowledge misappropriation hazard. (Our *post hoc* analyses provide some support for these perspectives.) Consider next how increasing interfirm modularity in the presence of higher outsourcee ignorance (cell C) balances these tradeoffs and enhances alliance performance without imposing knowledge sharing costs.

Modularization separates *what* the outsourced project does from *how* it accomplishes that action. Once the *what* aspect is prespecified in detail and kept invariable, the outsourcee has considerable leeway in *how* it completes project activities (such as concept development, detailed design, programming, and testing). Modularization ‘encapsulates’ the outsourcer’s private knowledge in the form of information about the interdependencies and interface specifications between the outsourced project and its technological portfolio—and nothing more (Parnas, 1972). The outsourced system can be designed to interoperate with the outsourcer’s technological portfolio without requiring detailed knowledge of the software code in other interdependent systems or the outsourcing firm’s private knowledge. Such encapsulation allows the alliance partners to manage interdependencies across firm boundaries without requiring the outsourcer to

share private knowledge with the outsourcee. Modularization thus creates an *embedded coordination mechanism* (Mikkola and Gassman, 2003; Sanchez and Mahoney, 1996). Superior alliance performance can thus be achieved without reducing outsourcee ignorance by increasing interfirm modularity. Therefore, interfirm modularity complements outsourcee ignorance. This leads to the following hypothesis, which proposes a positive interaction effect.

An increase in interfirm modularity enhances alliance performance as outsourcee ignorance of the outsourcing firm’s private knowledge increases.

To summarize, our key thesis is that increasing interfirm modularity facilitates successful accomplishment of the alliance objectives without requiring the outsourcer to lower outsourcee ignorance.

METHODS

Data collection

Data on 209 outsourcing alliances were collected during 2002–2003 from two informants for each project: (1) alliance liaison managers in U.S. firms and (2) lead project managers in the software services firms. The outsourcee-outsourcer pairs that represent each matched pair data point in our sample are independent. We contacted 818 software development firms in the three largest global software consortia in Russia, Ireland, and India: Russian National Software Development Alliance, Irish Investment and Development Agency, and India’s National Association of Software and Service Companies. Data on 232 projects were collected from 232 software services organizations, representing a 28.4 percent response rate. The response rate breakdown was: Russia 33.5 percent (59/176); Ireland 29.5 percent (54/183); India 25.9 percent (119/459). Next, alliance performance assessments were collected from the primary alliance liaison manager in the outsourcing firm. The 209 alliances for which such matched pair data were collected were included in the analyses. T-tests comparing the first 50 and last 50 respondents revealed no significant differences (see Armstrong and Overton, 1977) in project characteristics (such as cost overrun [$T = 1.54$], project

complexity in person-months [$T = -1.57$]) or outsourcee firm characteristics (such as capability maturity model (CMM) certification levels [$T = -1.46$], outsourcee's business application domain knowledge [$T = 0.185$], outsourcee firm age [$T = -0.95$]), providing no evidence of nonresponse bias.

Measures

Whenever possible, preexisting measurement instruments were used. New measures were developed for modularity, outsourcee ignorance, and alliance performance in software outsourcing partnerships based on a review of the literature and refined through detailed interviews with project managers in 19 software firms and seven academic experts. The unit of analysis is the project dyad, since the level of ignorance and modularity within the same relationship dyad can vary across different projects. *Interfirm modularity* was measured using a six item, seven-point scale that assessed the coupling between the outsourced system and the outsourcer's technological portfolio, based on descriptions of the construct in the modularity literature (Mikkola and Gassman, 2003; Parnas, 1972; Sanchez and Mahoney, 1996; Schilling, 2000). *Outsourcee ignorance* was measured by reverse scoring four items that assessed the outsourcee's knowledge about the project's application problem domain, that is, the outsourcer's project business objectives, business routines, and business rules implemented in the system. *Alliance performance* assessed project outcomes in terms of system reliability, functionality, meeting planned objectives and requirements, and overall fit with outsourcer needs. All constructs exhibited discriminant validity, high item-to-construct loadings, as well as scale reliability. The questionnaire items, the responding organization, and the reliability coefficients for the key constructs are summarized in the Appendix.

Control variables

To account for alternative explanations for alliance performance, we controlled for *project characteristics* (architecture design effort, team size, duration), *firm characteristics* (outsourcee firm age, CMM level, nationality), and *alliance characteristics* (prior joint experience and contract structure). Among the project characteristics, (1) higher effort

invested by the outsourcee in the front-end conceptual design phase (*architecture design effort*¹) (MacCormack, Verganti, and Iansiti, 2001), (2) *outsourcer liaison team size*, and (3) *project duration* (in months) can influence performance. Among firm characteristics, outsourcee *firm age* (measured as 2004–year of founding), and the maturity of its software development capabilities as measured by its CMM certification level (range: 1–5) (*CMM level*) (Ethiraj *et al.*, 2005) can influence project outcomes. Two dummy variables *Ireland* and *Russia* were included for *outsourcee nationality*. Among alliance characteristics, we controlled for prior partner-specific experience (a dummy set to 1 if the project was a *first-time collaboration*) (Oxley and Sampson, 2004) and for fixed price versus time-and-materials (dummy = 1; 67.6% in the sample) *contract structure*, recognizing that the latter do not penalize the outsourcee for midcourse changes requested by the outsourcer (Ethiraj *et al.*, 2005). Of the outsourcees that had worked previously with the same firm on earlier projects (59%), the relationship averaged 3.6 years. The following additional control variables considered were nonsignificant and subsequently dropped from the analyses: sales dependency of the outsourcee on the outsourcer, the outsourcer firm's technical knowledge, interfirm tie strength, technological turbulence, project size measured in thousand lines of code (KLOCs), and use of software-based interfirm development coordination tools. Construct correlations and descriptive statistics are summarized in Table 1.

RESULTS

We used a three-step ordinary least squares (OLS) regression model after evaluating potential endogeneity bias using the two-stage least squares (2SLS) procedure (see Wooldridge [2003: 506]). Interfirm modularity is potentially endogenous in that it requires outsourcer effort and knowledge, which itself might be a function of prior experience with the outsourcee. To evaluate endogeneity

¹ This was measured following MacCormack *et al.* (2001) as the percentage of the total project person-hours that were spent on architecture design out of the total project hours spent by the outsourcee on the following project activities: project management, architecture design, development, and testing.

Table 1. Construct correlations and psychometric properties

Constructs	Mean SD		Construct correlations													
	1	2	3	4	5	6	7	8	9	10	11					
1. Architecture design effort (%)	20.7	8.47														
2. Outsourcer-liaison team size	6.3	11.4														
3. Project duration (months)	10.65	9.56	0.263**													
4. Outsourcer firm age	6.99	6.27	0.166*	0.074												
5. Outsourcer CMM level	2.60	1.06	0.293**	0.171	0.057											
6. Ireland ⁺	—	—	0.187*	0.147	0.057	0.069										
7. Russia ⁺	—	—	0.098	0.117	0.038	0.196	0.328**									
8. First-time collaboration ⁺	—	—	0.305**	0.212*	0.1	0.019	0.036	0.061								
9. Contract structure ⁺	—	—	0.017	0.318**	0.066	0.023	0.189*	0.108	0.105							
10. Interfirm modularity	4.31	1.51	0.102	0.174*	0.1	0.164	0.06	0.156	0.1	0.224**						
11. Outsourcer ignorance	4.93	0.94	0.051	0.059	0.063	0.005	0.002	0.034	0.021	0.044	0.04					
12. Alliance performance	5.33	0.86	0.075	0.154	0.014	0.082	0.106	0.277*	0.201	0.028	0.13	0.201				

** p < 0.01, * p < 0.05 one-tailed test; ⁺ Dummy variable.

bias, we assessed the model using a 2SLS regression procedure using *first-time collaboration* as an instrument. We found no evidence for endogeneity bias in this 2SLS test, following the procedure outlined by Wooldridge (2003: 507). Wooldridge cautions that 2SLS estimation is less efficient than OLS when the independent variables in the model are exogenous. He therefore recommends using OLS when endogeneity is not observed, since this substantially decreases standard errors in estimates. Following Wooldridge's (2003: 506) recommendation to use OLS rather than 2SLS in the absence of evidence of endogeneity bias, we therefore estimated the model using a three-step OLS regression model. However, this does not entirely rule out the possibility of endogeneity bias in the model, which we discuss in our limitations.

First, the control variables were introduced in the model (Step 1) followed by the main effects (Step 2). The hypothesis was tested in the third step by adding a residual-centered interaction term between the outsourcee ignorance and interfirm modularity. The simultaneous analysis of main effects and interaction terms distorts the partial coefficients for the main effects because they tend to be highly correlated with the interaction terms. Using the uncorrected interaction term expectedly had a high variance inflation factor (VIF) of 26.85, a problem that we corrected using Lance's (1988) residual centering technique. In this two-stage procedure, we first regressed the interaction term on its component parts. We then used the resulting residual instead of the interaction term in the regression equation. This approach reduces multicollinearity between the interaction term and main effects, yielding a regression coefficient for the cross-product term that can be directly interpreted as the effect of the interaction term on the dependent variable. Although Lance's residual centering procedure is widely used in prior work, caution is warranted in light of its recent criticism (see Echambadi *et al.*, 2006). As Table 2 shows, the VIF (range: 1.02 to 1.30 <4) and tolerance (range: 0.77 to 0.98 ≥0.25) (variability in the predictor that is *not* explained by other predictors; representing the inverse of VIF) are well within the acceptable range emphasized by Wooldridge (2003).

We used two-tailed tests for the control variables, which were in the expected direction and explained 12.9 percent of the variance in the model. The main effects were nonsignificant, suggesting that in isolation outsourcee ignorance and

Table 2. Effects of interfirm modularity and outsourcee ignorance on alliance performance

	Step 1: Control variables				Step 2: Main effects				Step 3: Interaction effect			
	Regression		Collinearity statistics		Regression		Collinearity statistics		Regression		Collinearity statistics	
	β	T-statistic	Tolerance	VIF	β	T-statistic	Tolerance	VIF	β	T-statistic	Tolerance	VIF
Constant		11.66				8.37				8.80		
Architecture design effort	0.17	1.17	0.86	1.16	0.17	1.17	0.85	1.17	0.18	1.26	0.85	1.17
Outsourcee-liaison team size	0.25	1.60	0.77	1.29	0.26⁺	1.69	0.77	1.30	0.27⁺	1.83	0.77	1.30
Project duration	-0.23	-1.51	0.80	1.25	-0.23	-1.48	0.78	1.28	-0.22	-1.47	0.78	1.28
Outsourcee firm age	-0.05	-0.35	0.96	1.04	-0.07	-0.51	0.95	1.05	-0.08	-0.59	0.95	1.05
Outsourcee CMM level	-0.21	-1.38	0.83	1.21	-0.24	-1.56	0.81	1.23	-0.28⁺	-1.93	0.79	1.26
Ireland	-0.28⁺	-1.80	0.79	1.26	-0.28⁺	-1.81	0.79	1.27	-0.32[*]	-2.19	0.77	1.30
Russia	-0.40[*]	-2.61	0.83	1.21	-0.38^{**}	-2.50	0.80	1.25	-0.39^{**}	-2.71	0.80	1.25
First-time collaboration	-0.26⁺	-1.73	0.82	1.22	-0.25	-1.63	0.82	1.22	-0.22	-1.52	0.81	1.23
Contract structure	0.03	0.17	0.86	1.16	0.07	0.45	0.82	1.22	0.09	0.65	0.81	1.23
Interfirm modularity					-0.14	-0.97	0.86	1.16	-0.15	-1.09	0.86	1.16
Outsourcee ignorance					-0.20	-1.44	0.98	1.02	-0.19	-1.46	0.98	1.02
Outsourcee ignorance × Interfirm modularity									0.28[*]	2.08	0.95	1.06
R ² (Model F)	30.0% ⁺	(1.76)			35.4% ⁺	(1.74)			42.6%[*]	(2.10)		
R ² _{Adj}	12.9%				15.0%				22.4%			
ΔR^2_{Adj} (F-change)	—				2.1%(1.46)				7.4%^{***}	(4.32)		

***p < 0.001, **p < 0.01, *p < 0.05 two-tailed test; + p < 0.05 one-tailed test (significant results in bold); VIF = variance inflation factor; N = 209 alliances.

interfirm modularity are inconsequential to alliance performance. The relationship between modularity and performance is therefore a contingent one, consistent with the modularity literature (see Sanchez, 1995). Similarly, outsourcee ignorance does not by itself affect alliance performance. The positive, significant path coefficient (using a two-tailed test) and a statistically significant ΔR^2 associated with the addition of the interaction term (Table 2, Step 3) supports the complementarity hypothesis.

Post hoc analyses

To further explore how the results fit with the ideas summarized in Figure 1, we created an interaction plot (Figure 2) using high (+2 SD; dotted line) and low (-2 SD; solid line) levels of interfirm modularity. Overall, Figure 2 shows that higher outsourcee ignorance enhances alliance performance in higher modularity projects but decreases it in lower modularity projects. (Using ± 1 standard deviations yielded a similar but visually less pronounced pattern.) The pattern of relationships in Figure 2 therefore supports the subtleties discussed in our theory section.

The solid line in Figure 2 illustrates that under conditions of low modularity, an increase in outsourcee ignorance is associated with a decrease in alliance performance. The low ignorance side (left) of this plot suggests that a decrease in ignorance increases alliance performance in low modularity projects, supporting our argument regarding cell

A in Figure 1. The high ignorance side (right) of this plot further illustrates that an increase in ignorance in low modularity projects lowers alliance performance, consistent with our argument regarding cell D.

The dotted line in Figure 2 illustrates that under higher levels of modularity, an increase in outsourcee ignorance is associated with an increase in alliance performance. The low ignorance side (left) of this plot shows that a decrease in ignorance in high modularity projects is associated with a decrease in alliance performance, consistent with our argument regarding cell B in Figure 1 where the outsourcer potentially incurs both modularization costs and knowledge sharing costs. Although we had suggested that this might also exact a cost penalty on the outsourcer, a *post hoc* test of the relationship between outsourcee ignorance and project cost overruns (percentage by which the project exceeded its planned budget) was non-significant ($\beta = -0.7$, T-value -0.605), failing to support this assertion. The high ignorance side (right) of this plot shows that an increase in ignorance in high modularity projects is associated with an increase in alliance performance, consistent with cell C and our main hypothesis.

We also explored other performance benefits of modularity in addition to *post hoc* tests. Sanchez (1995: 150) has previously asserted that greater modularity also increases the speed of technology development activities, which we assessed in an additional *post hoc* test. In the software outsourcing domain, this is measured as an extent

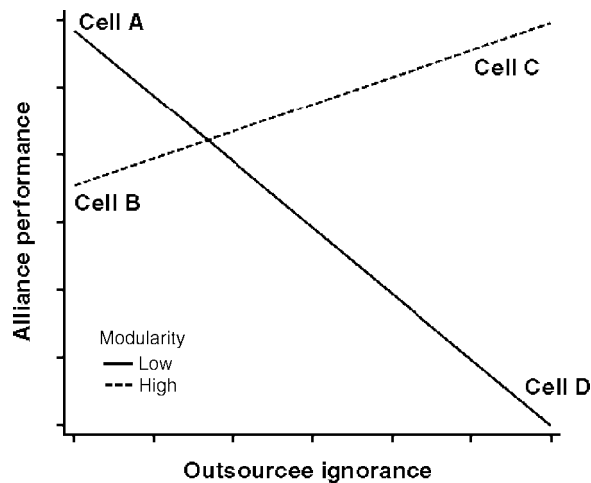


Figure 2. The relationships between alliance performance and outsourcee ignorance under high (+2 standard deviations) and low (-2 standard deviations) modularity

of schedule slippage (Ethiraj *et al.*, 2005). Using the percentage by which the project had overrun (or underrun) the original schedule as the dependent variable, we found that outsourcee ignorance ($\beta = 0.096$, T-value 0.976) and modularity ($\beta = 0.092$, T-value 0.94) had no significant effect, but their interaction term was negatively and significantly related with schedule slippage ($\beta = -0.191$, T-value -1.943). The interaction term added a modest 3.6 percent to the explained variance in schedule slippage but the change was significant (F-change 3.77, $p < 0.001$). Thus, contrary to Sanchez's (1995) simpler assertion, it is the *combination* of modular architectures with outsourcee ignorance that facilitates speedier development. More importantly, this suggests for future theory development that modularity coupled with ignorance might have benefits beyond enhancing alliance performance, such as enhancing alliance agility.

Two limitations should be noted in interpreting the results. First, although our limited tests failed to find evidence for endogeneity bias, both modularity and ignorance might be correlated with other unobserved qualities of the project that are not captured in our model. The results might possibly be affected by endogeneity bias. Second, although software outsourcing shares characteristics of other prevalent forms of knowledge-intensive outsourcing (e.g., call center operations, financial analysis, engineering design, and medical diagnosis), appropriate caution is warranted in generalizing these results to other types of alliances.

DISCUSSION AND IMPLICATIONS

We explored the theoretical assertion that interfirm modularity complements outsourcee ignorance in knowledge-intensive alliances. Although the modularity and alliances literatures imply that modularity lowers the need for interfirm knowledge sharing, the assertion has neither been theoretically developed nor tested. Analyses of data from 209 alliances between 209 Russian, Irish, and Indian software services firms and their U.S. clients show that interfirm modularity and outsourcee ignorance are complements. This insight into the interplay between modularity, ignorance, and alliance performance has important implications for research on alliances, KBV, and modularity.

Implications for research on strategic alliances

Alliances that seek to combine complementary resources and expertise from partnering firms create a tension between sharing enough private knowledge to successfully accomplish alliance goals and simultaneously safeguarding it against misappropriation. Our results suggest that increasing modularity at the project level complements outsourcee ignorance. Put another way, interfirm modularity lowers the need for interfirm knowledge sharing. Modularization thus encapsulates the outsourcer's private knowledge as modular project specifications that help manage interfirm interdependencies, enhancing alliance performance without requiring extensive knowledge sharing. This finding contributes modularity as a knowledge protection mechanism in alliances, which complements two other previously recognized ones vis-à-vis alliance governance (Gulati and Singh, 1998; Kale *et al.*, 2000) and scope reduction (Oxley and Sampson, 2004).

Implications for research on KBV

Although outsourcing shifts more production activities outside firm boundaries, modularization shifts more of the design responsibility in-house. This has a paradoxical implication for KBV: modularization requires that firms be able to predict *in advance* how the outsourced project will interact and integrate with its technological portfolio, which requires it to deepen its knowledge outside its own core domain. The ability to intentionally partition a complex technological portfolio into modular subsystems without losing critical information requires a broader architectural knowledge base in the outsourcing firm (Sanchez and Mahoney, 1996). Paradoxically, increasing specialization by outsourcing noncore activities might therefore require knowledge *despecialization* in the outsourcing firm. Thus seemingly 'peripheral knowledge' that facilitates alliance control (Tiwana and Keil, 2007) also facilitates interfirm modularization. This contributes directly to an emerging research stream on why firms sometimes 'need to know more than they make' (Brusoni *et al.*, 2001; Takeishi, 2002; Tiwana and Keil, 2007). This perspective also complements the fledgling literature on 'systems integration capability'—the ability of a firm to coordinate a network of specialist firms using capabilities broader than

the range of its in-house activities (Brusoni and Prencipe, 2001). Brusoni and Prencipe (2001) have recently observed such system integrators in the aircraft engine and chemical engineering industries. IKEA, Dell, and Li & Fung arguably represent other examples of system integrators that intentionally modularize their product designs and then outsource their production to a network of specialized partner firms. Finally, our emphasis on ignorance rather than knowledge raises a theoretically provocative question in KBV about when lack of knowledge itself can be strategically valuable in alliances. Ignorance, once lost, cannot readily be regained.

Implications for research on modularity

Both modularization and ignorance reduction are costly for an outsourcer. Modularization costs include the greater up-front design and specification effort to minimize interfirm interdependencies, along with the cost of maintaining peripheral knowledge to do so. Similarly, interfirm ignorance reduction can be time consuming, costly, and exacerbate knowledge misappropriation hazards. The outsourcer therefore faces a tension between when it should incur the costs of increasing modularity to gain the benefits of keeping the outsourcee relatively ignorant versus the costs of reducing outsourcee ignorance. We conjecture that the former strategy is preferable when the knowledge to be embodied in the outsourced project is rare, imitable, and competitively valuable to the outsourcing firm.

The results raise four fertile questions for future research. How does the adoption of modular outsourcing influence the divergence between 'what a firm does' and 'what a firm knows' that recent research has highlighted (Brusoni *et al.*, 2001; Takeishi, 2002; Tiwana and Keil, 2007)? Do formal and informal alliance governance mechanisms complement or substitute for interfirm modularity? Does interfirm modularity lower interfirm knowledge spillovers in both directions? How can firms manage the trade-offs between the potential loss of integrality and the benefits of modular subcontracting to specialized alliance partners with potentially greater expertise and scale advantages?

The overarching theoretical implication is that from the outsourcer's perspective, outsourcee ignorance is bliss but only under the right conditions. A firm does not *have* to—but sometimes has

strong incentives to—rely on interfirm ignorance in alliances. When it does so, it is important to match that ignorance with the appropriate level of modularity. Paradoxically, this implies that specialized technical knowledge that is peripheral to the outsourcing firm, and that it seeks to access using the alliance, might itself be necessary for increasing interfirm modularity. More broadly, the study raises the provocative question about when greater ignorance—rather than knowledge—is valuable in alliances.

ACKNOWLEDGEMENTS

The developmental feedback from Editor Ed Zajac and the anonymous *SMJ* reviewers is gratefully acknowledged. Inputs from academic colleagues including Ashley Bush, Mark Keil, Bala Ramesh, and Benn Konsynski are also gratefully acknowledged.

REFERENCES

- Adelson B, Soloway E. 1985. The role of domain experience in software design. *IEEE Transactions on Software Engineering* **11**(11): 1351–1360.
- Armstrong J, Overton T. 1977. Estimating nonresponse bias in mail surveys. *Journal of Marketing Research* **14**(3): 396–402.
- Becker GS, Murphy KM. 1992. The division of labor, coordination costs, and knowledge. *Quarterly Journal of Economics* **107**(4): 1137–1160.
- Brusoni S, Prencipe A. 2001. Unpacking the black box of modularity: technologies, products and organizations. *Industrial and Corporate Change* **10**(1): 179–205.
- Brusoni S, Prencipe A, Pavitt K. 2001. Knowledge specialization, organizational coupling, and boundaries of the firm: why do firms know more than they make? *Administrative Science Quarterly* **46**: 597–621.
- Echambadi R, Arroniz I, Reinartz W, Lee J. 2006. Empirical generalizations from brand extension research. *International Journal of Research in Marketing* **23**(6): 253–261.
- Ethiraj SK, Kale P, Krishnan MS, Singh JV. 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.
- Gulati R, Singh H. 1998. The architecture of cooperation: Managing coordination costs and appropriation concerns in strategic alliances. *Administrative Science Quarterly* **43**: 781–814.
- Hoopes DG, Postrel S. 1999. Shared knowledge, 'glitches,' and product development performance. *Strategic Management Journal* **20**(9): 837–865.
- Kale P, Singh H, Perlmutter H. 2000. Learning and protection of proprietary assets in strategic alliances:

- building relational capital. *Strategic Management Journal*, March Special Issue **21**: 217–237.
- 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.
- MacCormack A, Verganti R, Iansiti M. 2001. Developing products on internet time: the anatomy of a flexible development process. *Management Science* **47**(1): 133–150.
- Matusik SF. 2002. An empirical investigation of firm public and private knowledge. *Strategic Management Journal* **23**(5): 457–467.
- Messerschmitt D, Szyperski C. 2003. *Software ecosystem*. MIT Press: Cambridge, MA.
- Mikkola J, Gassman O. 2003. Managing modularity of product architectures: toward an integrated theory. *IEEE Transactions on Engineering Management* **50**(2): 204–218.
- 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.
- Prahalad C, Krishnan M. 2002. The dynamic synchronization of strategy and information technology. *Sloan Management Review* **43**(4): 24–33.
- Robillard P. 1999. The role of knowledge in software development. *Communications of the ACM* **42**(1): 87–92.
- 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.
- 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.
- Takeishi A. 2002. Knowledge partitioning in the interfirm division of labor: the case of automotive product development. *Organization Science* **13**(3): 321–338.
- Tiwana A, Keil M. 2007. Does peripheral knowledge complement control? An empirical test in technology outsourcing alliances. *Strategic Management Journal* **28**(6): 623–634.
- Wooldridge J. 2003. *Introductory Econometrics: A Modern Approach*. South-Western College Publishing: Mason, Ohio.

APPENDIX: CONSTRUCT MEASURES

Interfirm modularity ($\alpha = 0.72$; seven-point semantic differential scale)

How would you characterize the relationship between this system and the systems in the client organization with which this system had to interoperate? (1) Tightly coupled—Loosely coupled, (2) Evolving interface specifications—Stable interface specifications, (3) Integrated—Modular, (4) Interdependent—Independent, (5) Customized—Plug-and-play, (6) Changes in this system affect interoperability with client's other systems—Changes in this system do not affect interoperability with client's other systems.

Alliance performance ($\alpha = 0.91$; scale anchors: 1 = *Much Worse* and 7 = *Much Better*)

Compared to other IT projects completed by your company, how would you characterize this

project's outcomes? (1) System reliability, (2) implementation of functionality, (3) meeting project objectives, (4) meeting functional requirements, and (5) overall fit with our needs.

Outsourcee ignorance ($\alpha = 0.82$)

The outsourcee firm's knowledge in the following areas specific to the project: (a) the outsourcing firm's business processes, (b) the outsourcing firm's business objectives, (c) the outsourcing firm's day-to-day business routines, (d) the business rules implemented in this system. This scale was reverse scored to compute ignorance.