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 ‘outsourcer’ 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 Gerdin 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 outsourcer 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 outsourcer 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 outsourcers. 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 strategic value of ignorance rather than 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
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 outsourcer 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 \( T = -1.46 \)), outsourcer’s business application domain knowledge \( T = 0.185 \), outsourcer 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, outsourcer 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). **Outsourcer ignorance** was measured by reverse scoring four items that assessed the outsourcer’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 (outsourcer firm age, CMM level, nationality), and alliance characteristics (prior joint experience and contract structure). Among the project characteristics, (1) higher effort invested by the outsourcer 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, outsourcer 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 outsourcer 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 outsourcer 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 outsourcer 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 outsourcer. 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 outsourcer on the following project activities: project management, architecture design, development, and testing.
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

<table>
<thead>
<tr>
<th>Step 1: Control variables</th>
<th>Step 2: Main effects</th>
<th>Step 3: Interaction effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression Collinearity statistics</td>
<td>Regression Collinearity statistics</td>
<td>Regression Collinearity statistics</td>
</tr>
<tr>
<td>β T-statistic Tolerance VIF</td>
<td>β T-statistic Tolerance VIF</td>
<td>β T-statistic Tolerance VIF</td>
</tr>
<tr>
<td>Constant</td>
<td>11.66</td>
<td>8.37</td>
</tr>
<tr>
<td>Architecture design effort</td>
<td>0.17</td>
<td>0.17</td>
</tr>
<tr>
<td>Outsourcer-liaison team size</td>
<td>0.25</td>
<td>0.26**</td>
</tr>
<tr>
<td>Project duration</td>
<td>−0.23</td>
<td>−0.23</td>
</tr>
<tr>
<td>Outsourcer firm age</td>
<td>−0.05</td>
<td>−0.07</td>
</tr>
<tr>
<td>Outsourcer CMM level</td>
<td>−0.21</td>
<td>−0.24</td>
</tr>
<tr>
<td>Ireland</td>
<td>−0.28**</td>
<td>−0.28**</td>
</tr>
<tr>
<td>Russia</td>
<td>−0.40**</td>
<td>−0.38**</td>
</tr>
<tr>
<td>First-time collaboration</td>
<td>−0.26**</td>
<td>−0.26**</td>
</tr>
<tr>
<td>Contract structure</td>
<td>0.03</td>
<td>0.07</td>
</tr>
<tr>
<td>Interfirm modularity</td>
<td>−0.25</td>
<td>−0.25</td>
</tr>
<tr>
<td>Outsourcee ignorance</td>
<td>−0.14</td>
<td>−0.19</td>
</tr>
<tr>
<td>Outsourcee ignorance × Interfirm modularity</td>
<td>−0.14</td>
<td>−0.14</td>
</tr>
<tr>
<td>R² (Model F)</td>
<td>30.0%*</td>
<td>35.4%*</td>
</tr>
<tr>
<td>R²Adj</td>
<td>12.9%</td>
<td>15.0%</td>
</tr>
<tr>
<td>ΔR²Adj (F-change)</td>
<td>2.1% (1.46)</td>
<td>7.4%** (4.32)</td>
</tr>
</tbody>
</table>

*** 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 $\Delta R^2$ 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 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 additional 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...
of schedule slippage (Ethiraj et al., 2005). Using the percentage by which the project had over-
rung or underrun) the original schedule as the
dependent variable, we found that outsourcee igno-
rance \((\beta = 0.096, \text{T-value } 0.976)\) and modular-
ity \((\beta = 0.092, \text{T-value } 0.94)\) had no significant
effect, but their interaction term was negatively
and significantly related with schedule slippage
\((\beta = -0.191, \text{T-value } -1.943)\). The interaction
term added a modest 3.6 percent to the explained
variance in schedule slippage but the change was
significant \((F\text{-change } 3.77, p < 0.001)\). Thus, con-
trary to Sanchez’s (1995) simpler assertion, it is
the combination of modular architectures with out-
sourcee ignorance that facilitates speedier devel-
opment. 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 interpret-
ing 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. Sec-
ond, although software outsourcing shares char-
acteristics of other prevalent forms of knowledge-
intensive outsourcing (e.g., call center operations,
financial analysis, engineering design, and med-
ical 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 mod-
ularity and alliances literatures imply that mod-
ularity 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 per-
formance 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 cre-
ate a tension between sharing enough private
knowledge to successfully accomplish alliance
goals and simultaneously safeguarding it against
misappropriation. Our results suggest that increas-
ing modularity at the project level complements
outsourcee ignorance. Put another way, interfirm
modularity lowers the need for interfirm knowl-
dge 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 knowl-
dge 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 activ-
ities outside firm boundaries, modularization shifts
more of the design responsibility in-house. This
has a paradoxical implication for KBV: modular-
ization requires that firms be able to predict in
advance how the outsourced project will inter-
act and integrate with its technological portfolio,
which requires it to deepen its knowledge outside
its own core domain. The ability to intention-
ally partition a complex technological portfolio
into modular subsystems without losing critical
information requires a broader architectural knowl-
dge base in the outsourcing firm (Sanchez and
Mahoney, 1996). Paradoxically, increasing spe-
cialization by outsourcing noncore activities might
therefore require knowledge despecialization in
the outsourcing firm. Thus seemingly ‘periph-
eral 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 capabil-
ity’—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

Kale P, Singh H, Perlmutter H. 2000. Learning and protection of proprietary assets in strategic alliances:


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.

**Outsourcer ignorance** ($\alpha = 0.82$)

The outsourcer 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.