

Association for Information Systems AIS Electronic Library (AISeL)

ICIS 2002 Proceedings

International Conference on Information Systems
(ICIS)

December 2002

How Good Are These UML Diagrams? An Empirical Test of the Wand and Weber Good Decomposition Model

Andrew Burton-Jones
Georgia State University

Peter Meso
Georgia State University

Follow this and additional works at: <http://aisel.aisnet.org/icis2002>

Recommended Citation

Burton-Jones, Andrew and Meso, Peter, "How Good Are These UML Diagrams? An Empirical Test of the Wand and Weber Good Decomposition Model" (2002). *ICIS 2002 Proceedings*. 10.
<http://aisel.aisnet.org/icis2002/10>

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

HOW GOOD ARE THESE UML DIAGRAMS? AN EMPIRICAL TEST OF THE WAND AND WEBER GOOD DECOMPOSITION MODEL

Andrew Burton-Jones

J. Mack Robinson College of Business
Georgia State University
Atlanta, GA USA
abjones@cis.gsu.edu

Peter Meso

J. Mack Robinson College of Business
Georgia State University
Atlanta, GA USA
pmeso@cis.gsu.edu

Abstract

In 1989, Wand and Weber proposed a formal model of systems decomposition based on ontology. Chidamber and Kemerer (1994) soon applied this model to develop complexity metrics for object-oriented design (OOD). Chidamber and Kemerer's OOD metrics suite continues to receive interest in software engineering (Bansiya and Davis 2002; Basili et al. 1996). To date, however, Wand and Weber's good decomposition model has received almost no application in information systems (IS). For three reasons, we believe the theory might assist IS researchers. First, object-oriented analysis (OOA) has not been as successful in practice as OOD or OO programming (Chuang and Yadav 2000). The good decomposition model may help IS researchers investigate improvements to OOA. Second, Johnson (2002) recently lamented how few OOA studies employ any theory. Wand and Weber's theory may, therefore, be a useful approach. Third, many believe OOA is a revolutionary step away from traditional approaches (Sircar et al. 2001). Practicing analysts could benefit from theory-based principles to guide their use of this "revolutionary" technique.

In this study, we report an experiment to determine the utility of the good decomposition model in OOA. We operationalized each condition of Wand and Weber's model in a set of UML diagrams and tested participants' understanding of the diagrams across three levels. Our results lend support to Wand and Weber's theory, but only across dependent variables that tested participants' actual understanding. The impact on participants' perceptions of their understanding remained equivocal.

Keywords: Object-oriented analysis, unified modeling language, UML, ontology, decomposition

1 INTRODUCTION

When systems analysts specify requirements for new applications, they often use diagrams called conceptual models. These diagrams are designed to support communication between developers and users, to help analysts understand a domain, and provide an input to systems design (Wand and Weber 2002). Following the move toward object-oriented (OO) development in industry, at least 19 OO modeling specifications have emerged (Wieringa 1998). The UML (unified modeling language) has been adopted as the standard by the Object Management Group (Kobryn 1999). The aim of this study is to empirically test a theory that might provide a basis for improving OO analysis (OOA) in UML.

To date, most OOA research has compared OO to traditional approaches and studied relative strengths and weaknesses (e.g., Agarwal et al. 2000; Hardgrave and Dalal 1995; Vessey and Conger 1994). There has been less focus to date on improving use of any one method on its own. The standardization of UML, however, affords an opportunity to focus on improving the "science" of diagramming in this suite. Given the revolutionary nature of object-oriented analysis (OOA) (Sircar et al. 2001) and the lack of theory in studies to date (Johnson 2002), theory-driven approaches to improving UML diagramming are essential.

The objective of this paper is to empirically test the usefulness of Wand and Weber’s good decomposition model as a foundation for improving OOA in UML. The good decomposition model is a set of formal conditions that define characteristics of well-specified systems (Weber 1997). The conditions are based on an ontology (or meta-theory of the structure and behavior of real world systems) proposed by Bunge (1977). We recognize upfront that the good decomposition model can only provide a partial solution for OOA. The model primarily operates at a *semantic* level and scholars recognize that syntactic, pragmatic, and social concerns are also important (Shanks 1999). Our objective is to test the benefit of this semantic level theory.

We focus on the good decomposition model for two reasons. First, it is the only theory we know of that (1) provides a formal set of properties of “good” decompositions and hence is clearly refutable, and (2) covers all aspects of systems modeling and can thus be used right across the UML suite. Second, while there have been no tests (to our knowledge) of the good decomposition model in systems analysis, the model formed a foundation of Chidamber and Kemerer’s (1994) metrics suite in OO design, which has had ongoing impact in software engineering (Citeseer reported 141 citations by March 2002). The model also proved promising in an early study of non-OO design (Paulson and Wand 1992). Like Basili et al.’s (1996) test of Chidamber and Kemerer’s metric suite, our purpose is to empirically validate the good decomposition model to determine its usefulness for OOA. Specifically, we want to test the following research question: *Do UML diagrams that manifest better decompositions increase analysts’ understanding of a domain?*

2 THE GOOD DECOMPOSITION MODEL

In systems development, decomposition involves breaking down a high-level statement of requirements into smaller pieces to be designed and implemented. Better decompositions are more likely to result in systems that can be understood, maintained, and perform effectively (Parnas 1972). There are many *levels* to an IS decomposition, from an initial statement of requirements to implemented code. This paper tests the utility of Wand and Weber’s theory for assisting an important early level: the decomposition from the initial statement of requirements to an OOA specification of those requirements in UML (see Figure 1).

As Figure 1 shows, an analyst could propose many alternative decompositions to represent a real world domain. The good decomposition model provides a set of metrics for determining the relative quality of these decompositions. It is important to note that an analyst could also develop multiple decompositions of equal quality (whether good or bad) (Paulson and Wand 1992). The good decomposition model thus does not assume that there is one best way to model the real world. The model is also deliberately parsimonious. Other metrics can be used in combination with the model to compare “equivalent” decompositions (see Paulson and Wand 1992; Chidamber and Kemerer 1994). This paper is limited to a test of the good decomposition model alone.

The good decomposition model proposes five conditions required for a good decomposition: *minimality*, *determinism*, *losslessness*, *minimum coupling*, and *strong cohesion* (Weber 1997) (described in Table 1).

Minimality requires that a system does not contain any redundant state variables (attributes). In a UML diagram, for instance, redundant state variables would include attributes that are not used by any method in the system and are unnecessary for the system to perform its function.

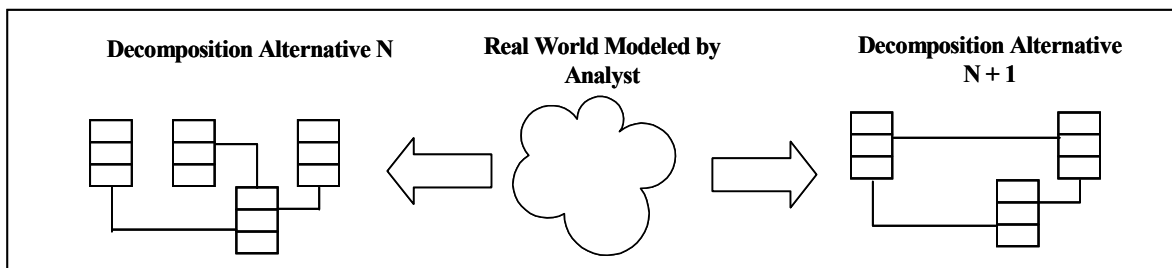


Figure 1. Alternative Decompositions from the Real World to a Conceptual Model

Table 1. Summary of Wand and Weber Good Decomposition Conditions

Condition	Description (adapted from Weber 1997; cohesion condition also in Dromney 1996)
Minimality	A decomposition is considered good only if for every subsystem at every level in the level structure of the system there are no redundant state variables.
Determinism	For a given set of external (input) events at the system level, a decomposition is good only if for every subsystem at every level in the level structure of the system an event is either an external event or a well-defined internal event.
Losslessness	A decomposition is good only if every hereditary state variable and every emergent state variable in a system is preserved in the decomposition.
Minimum Coupling	A decomposition has minimum coupling <i>iff</i> the cardinality of the totality of input for each subsystem of the decomposition is less than or equal to the cardinality of the totality of input for each equivalent subsystem in the equivalent decomposition.
Strong Cohesion	A set of outputs is maximally cohesive if all output variables affected by input variables are contained in the same set, and the addition of any other output to the set does not extend the set of inputs on which the existing outputs depend.

Determinism relates to system dynamics. *External* events are changes of state in the focal system caused by a change of state in another system or subsystem. Such events are inherently unpredictable (e.g., a class cannot know when another class will send it a message or its message contents). While unpredictable, the determinism condition requires that external events still be named so the system knows their existence and source. *Internal* events are internal state changes controlled by the system. For example, methods in a class can operate on attributes in their class. Determinism requires that all internal events be named and well defined. Internal events are well defined when, given a specific pre-event state, an internal event leads predictably to one and only one post-event state.

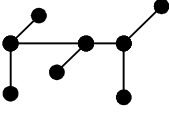
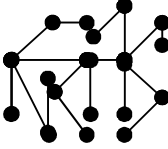
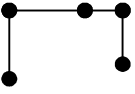
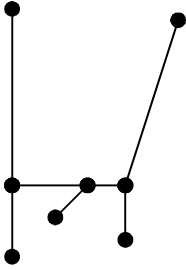
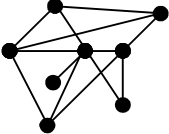
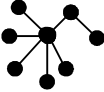
Losslessness requires that emergent properties (or properties of higher-level systems) are not lost during decomposition. A property of a customer relationship management system, for instance, might be customer satisfaction. As the system is broken down into its constituent parts, the losslessness condition requires that the customer satisfaction property is not lost in the final system.

Minimum coupling is judged in the context of *equivalent* decompositions, *viz.*, ones that contain the same objects and components (Weber 1997). Coupling increases with interaction between subsystems. In OOD, coupling can be defined as a method in one object using methods or instance variables in another object (Chidamber and Kemerer 1994). Degree of coupling can then be defined as the number of other classes to which an object is coupled (including the number of ways that each object is coupled to any other object).

Strong cohesion in the good decomposition model is based on Dromney's (1996) definitions. Chidamber and Kemerer applied Wand and Weber's definition in OOD as the similarity of methods in a class; the larger the number of similar methods, the more cohesive the class. For example, a class that contains attributes and methods that can logically be separated into two sets (i.e., are dissimilar) would violate Wand and Weber's and Chidamber and Kemerer's definition of strong cohesion.

To summarize, the good decomposition model provides a set of conditions proposed to be necessary for good system specification. Wand and Weber propose that analysts and users will be more able to understand systems complying with the conditions. Recall, however, that Wand and Weber's model derives from ontology (philosophy) (Bunge 1977). To predict that these conditions affect analysts' *understanding* of OOA scripts requires that the conditions be supported by theories of cognition. Table 2 summarizes how violations of each of these conditions reduce an individual's understanding based on theories of semantic memory (Collins and Quillan 1969). In semantic network theory, individuals store concepts in memory as nodes connected by paths (Ashcraft 2002). To perform cognitive activities, concepts from memory must be recalled. This follows a process of spreading activation. A node is primed in memory which leads to the paths connecting to it being activated. Activation has to be strong enough for a search along the path to reach a connected node. In these theories, understanding (typically operationalized as accuracy and response latency) is positively related to the strength of activation (Ashcraft 2002).

Table 2. Impact of Good Decomposition Violations on Human Understanding

Illustrations of Impact in Semantic Memory Caused by <i>Violations</i> of Good Decomposition:	
	
	
	
	
Violation	Impact on Semantic Memory (adapted from Weber 1997)
Minimality	Increases the number of nodes in memory. This diffuses the strength of priming of any one node and decreases the likelihood that a search in memory will be successful.
Losslessness	Nodes representing emergent properties are lost. A search for these nodes in memory will be unsuccessful.
Determinism	When events are not well specified, the state of other subsystems must be known before a subsystem's behavior can be understood. More distant nodes must be accessed. These nodes will experience weaker priming and retrieval will be slower and less accurate.
Minimum Coupling	As the strength of coupling between nodes increases, the number of paths between nodes increases. This dissipates the level of priming on any one path leading to higher response latency and less accuracy.
Strong Cohesion	If a subsystem lacks cohesion, the nodes connected directly to it will relate to multiple functions that the system performs. When an individual focuses on the node, all connected nodes will be primed but only some are relevant to understanding. Priming of <i>relevant</i> nodes is thus weaker so response latency (accuracy) will increase (decrease).

An assumption of these arguments, however, is that individuals encode elements (or chunks) of a conceptual model directly into memory in a more or less one-to-one mapping so that the violations in the model are also manifest in their semantic network. Theories of memory suggest two situations when this assumption might not hold. First, when individuals have plenty of time to analyze a model, they are likely to engage in elaboration processes to restructure their semantic network. Elaboration can improve an individual's memory by increasing the priming of nodes in memory and by improving the structure of the semantic network (Weber 1997). Second, when individuals are expert in the domain represented, they will already have an existing semantic network. Rather than encode the conceptual modeling constructs in a one-to-one fashion into memory, they can internalize the model by adjusting their existing semantic network. Moreover, where there are clear problems or ambiguities in the conceptual model, they can use their existing knowledge to determine a more plausible or correct interpretation of the terms or relationships in the model when internalizing it in memory (Burton-Jones and Weber 1999). The implication for empirically testing the good decomposition model is that the test should not allow participants significant time,¹ nor represent a domain in which the participants are expert. Empirical tests where these conditions do not hold may not find significant results (see Weber 2001; we thank an anonymous reviewer for calling our attention to this paper).

These arguments lead us to develop our formal propositions. We propose, subject to the assumptions above, that UML diagrams that manifest better decompositions will communicate meaning more effectively and thereby improve individuals' understanding. Further, we posit that violations of the good decomposition conditions have approximately equal weight on individuals' understanding, so that they operate in an additive (approximately linear) fashion. Table 3 states our four propositions. The next section outlines our experiment to test these propositions.

¹We note that there is no clear definition of a "significant time." Research by Siau and Tian (2001) may ultimately give us a better understanding of the time it takes to encode conceptual modeling constructs in memory, but this research is still ongoing.

Table 3. Propositions

#	Proposition
1	UML scripts complying with all five conditions (good decomposition) will be more understandable than UML scripts violating all conditions (bad decomposition)
2	UML scripts complying with all five conditions (good decomposition) will be more understandable than UML scripts violating three conditions (moderate decomposition)
3	UML scripts violating three conditions (moderate decomposition) will be more understandable than UML scripts violating all conditions (bad decomposition)
4	Better decompositions will be more understandable (the relationship is approximately linear)

3 METHOD

We used a laboratory experiment to test our propositions. Given the exploratory nature of our research, this method helped us to control for other factors that might impact users' understanding of UML diagrams.

3.1 Design

The experiment used a 1*3 between-groups design. Each group received three UML diagrams that either fully complied (good), partially complied (moderate), or violated all of the good decomposition conditions (bad). Participants were randomly assigned across groups. Order of tasks was randomly assigned to control for learning effects. The dependent variables were: (1) problem-solving performance; (2) performance in a cloze (or fill-in-the-blanks) test; and (3) perceived ease-of-understanding the diagrams.

3.2 Participants

A total of 59 senior-level students studying OOD at a large university in the southern part of the United States participated. Two responses could not be included (one subject arrived too late, the other misunderstood the instructions), resulting in three groups of 19 (N = 57). Participants were awarded credit but attendance was voluntary. All students took an OOA course as a prerequisite and both courses used UML. Participants reported that they first learned UML (on average) 6 months prior. Their average self-reported expertise in the domain modeled was 3.5 on a 7-point scale (where 4 represented the same as a practitioner). Overall, while participants were familiar with UML and the domain tested, they could not be considered expert in either.

3.3 Materials

Participants received three diagrams, a use case, class, and state transition diagram. The diagrams derived from a case adapted from Conger (1994). Three versions of each were created: good, moderate, and bad. The moderate decomposition violated *minimality*, *losslessness*, and *determinism*. These were selected because we considered them less familiar to the IS community than *coupling* and *cohesion*. The bad decomposition violated all conditions. Figures 2, 3, and 4, show the bad class, state transition, and use case diagrams respectively. To conserve space, we do not show the good or moderate diagrams, but have annotated the bad diagrams to explain the differences in the diagrams. All materials are available on request.

Prior research has not operationalized the good decomposition conditions in OOA. To manipulate each condition, we used definitions in Weber (1997), Chidamber and Kemerer (1994), and Parsons and Wand (1997). Each manipulation was made to specific aspects of the diagrams to isolate their effects as much as possible. To ensure validity, an independent academic expert on the topic reviewed the diagrams, and changes were made based on this feedback. As shown in Figure 2, *minimality* was manipulated by including redundant attributes in the applicant and job contract classes. *Losslessness* was manipulated in the 1..n relationships between the Applicant→Employee-Contract and Client→Job classes. The good diagram included overall performance and overall satisfaction ratings in the applicant and client classes. These emergent properties were lost in the bad

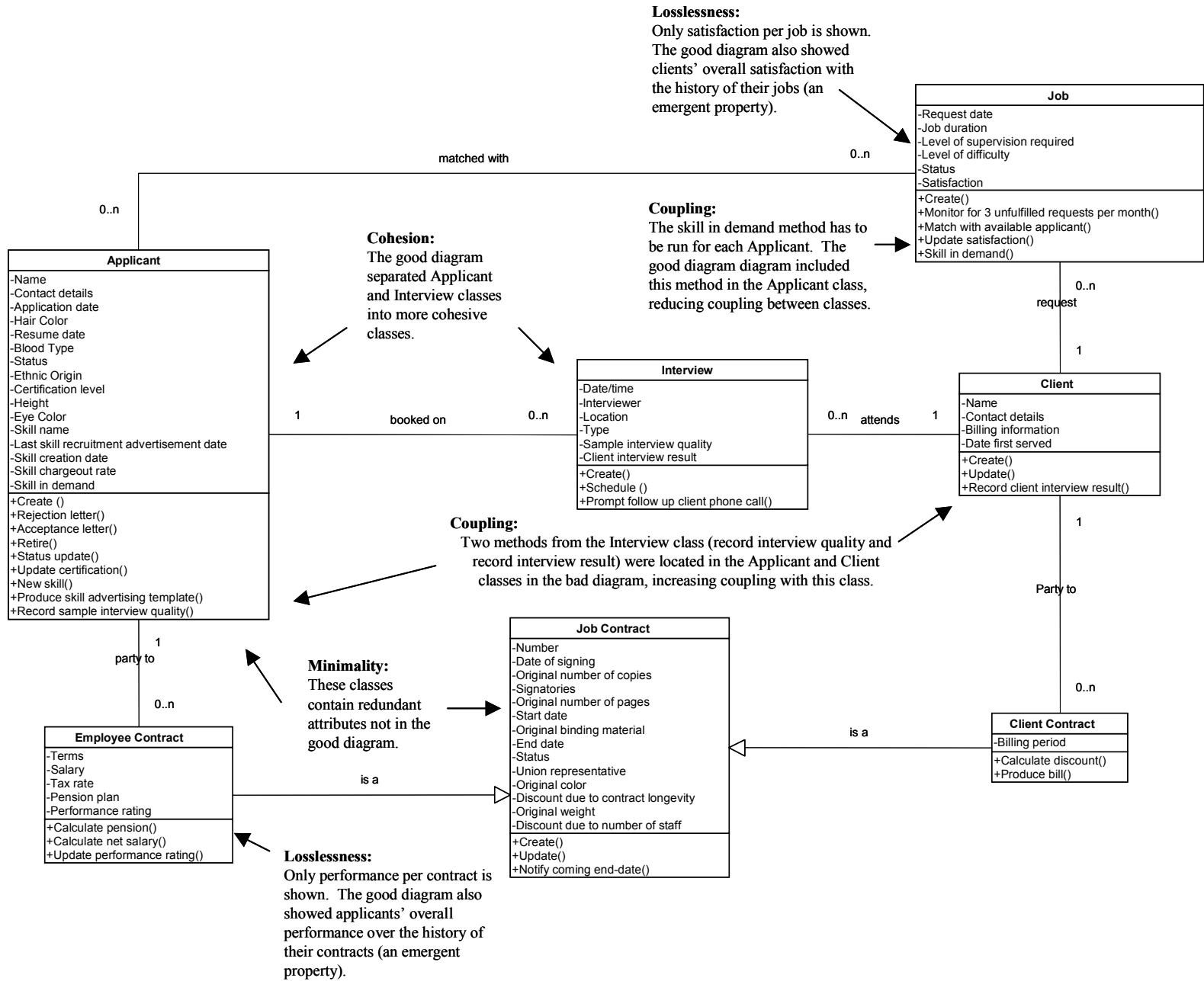
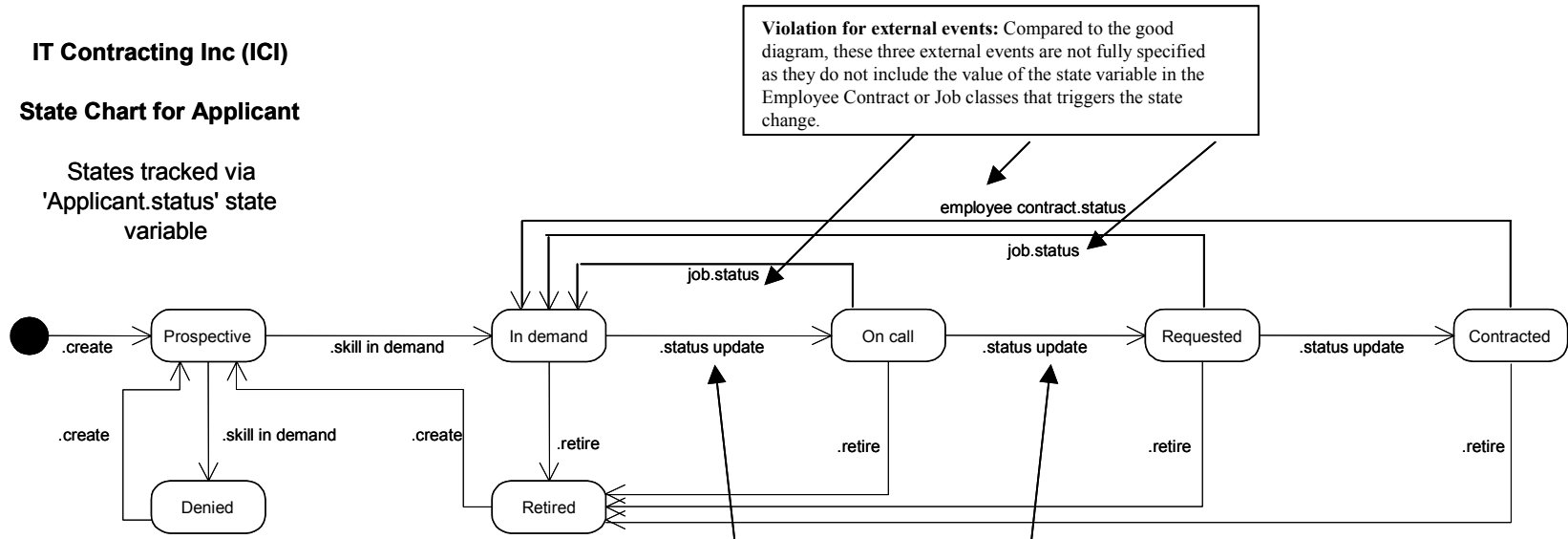


Figure 2. Class Diagram: Bad Decomposition

IT Contracting Inc (ICI)

State Chart for Applicant

States tracked via
'Applicant.status' state variable



Violation for external events: Compared to the good diagram, these three external events are not fully specified as they do not include the value of the state variable in the Employee Contract or Job classes that triggers the state change.

Violation for internal events: Compared to the good diagram, the *.status update* event here is indeterminate because for two states "On Call," and "Requested," the internal event leads to two further internal events (*.status update* and *.retire*). The good model showed that the transition between these states depended on state changes in classes other than Applicant.

Legend:

- Initial state
- State1 State
- Transition
- Internal event: Details the operation the Class performs to change its state.
- External event: Details the name of the external class and its state variable that triggers the change of state shown.

Figure 3. State Transition Diagram: Bad Decomposition

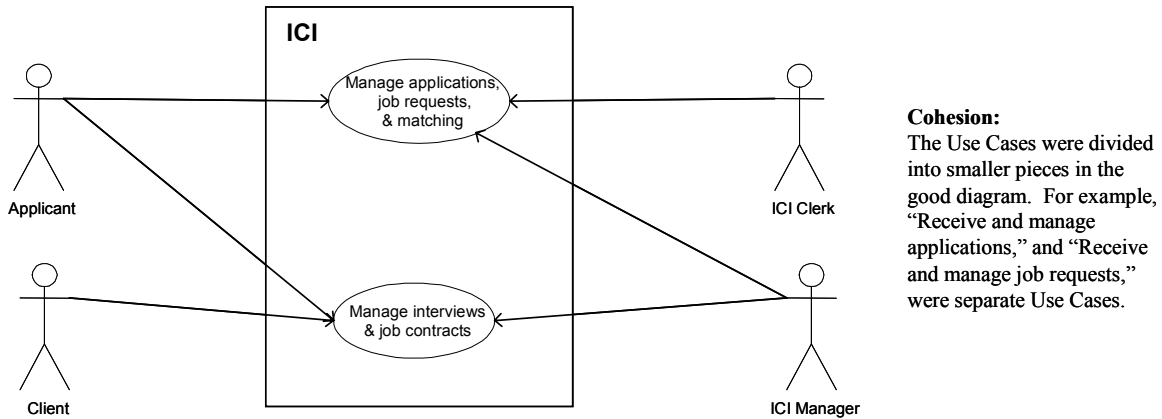


Figure 4. Use Case Diagram: Bad Decomposition

diagram where only the hereditary properties of job satisfaction and contract performance remained. *Determinism* was manipulated in the state transition diagram (Figure 3). The external events (“job status” and “employee contract status”) were under specified and the internal event status update was indeterminate because for two states (in demand and on call), *two* internal events emerged from the subsequent state. *Minimum coupling* was violated by methods that increased coordination between classes (“skill-in-demand” in the job rather than applicant class, and “record client interview result” and “record sample interview quality” in the client and applicant rather than interview class). *Strong cohesion* was violated in the use case (Figure 4) by using more aggregated use cases, and in the class diagram (Figure 2) by collapsing the service and applicant classes into one class and the sample and final interview subclasses into one class, reducing similarity of methods. Apart from the violations, the diagrams between groups were informationally equivalent (i.e., contained the same semantics) (per Kim et al. 2000).

3.4 Dependent Measures

Our dependent measures were based on education research that found that understanding is best tested by *deep* processing such as problem solving (Mayer 1989). Following Gemino (1998) and Bodart et al. (2001), the problem-solving measure in our research was participants’ number of acceptable answers to 11 problem-solving questions asked about the domain. Each question asked for a selection of answers to a relevant business problem and an explanation for each answer. Each answer-explanation pair was worth one mark (half a mark for each part). Requiring an explanation for each answer ensured that we tested deep rather than just surface understanding. For example, one problem-solving question was:

ICI has a number of applicants with skills in high demand but who are not yet contracted with a client. From the information provided in the models, list up to six possible causes for this and explain how each might have led to this situation.

We assessed participants’ answers by creating a set of “acceptable” answers to each question (Mayer 1989). For example, acceptable answers to the above question included: “low quality in sample interview,” “clients cancel jobs before contract starts,” “low certification,” and “low performance rating.” Of the 11 questions, the answers for two questions came from one diagram, the answers for six questions came from two diagrams, and the answers for four questions came from all three diagrams. The questions did not stipulate, however, which diagrams would be useful.

We also tested participants understanding via a fill-in-the-blanks test whereby the narrative was provided for participants to complete. We assessed participants based on the number of blanks they filled with a correct word or synonym. For example, an extract of the fill-in-the-blanks test was:

Applicants’ _____ is recorded as _____ when their applications are received. If their _____ are _____ they are moved to an _____ state. If they perform to _____ in the _____ interview they are moved to _____.

In accordance with the assumptions of our tests (outlined in section 2 above), the problem-solving and fill-in-the-blank tests were designed to be challenging but not unreasonable. Specifically, students were given approximately three minutes per problem-solving question and were required to complete 116 blanks or approximately 4.5 blanks per minute in the fill-in-the-blanks test. Students were advised to do the best they could but not to worry if they could not finish the experiment.

To measure participants' perceived ease-of-understanding their diagrams, we adapted Davis' (1989) perceived ease-of-use instrument. Four items were chosen from the instrument in Moore and Benbasat (1991) and Gemino (1998) and adapted to ask participants about their models' "ease-of-understanding" rather than "ease-of-use." For example, one item of the scale was:

Trying to understand the UML diagrams of ICI required a lot of mental effort.

An independent coder and one of the researchers separately coded the problem-solving and fill-in-the-blanks answers. Scoring the problem-solving answers required some subjectivity. The coding scheme appeared reliable, however, as the correlation between coders for the 11 questions ranged from $r = 0.87$ to $r = 0.97$. The fill-in-the-blanks test was much more objective, reflected in the correlation between coders ($r = 0.99$). The results reported in section 4 are based on the independent coder's scores for the problem-solving and fill-in-the-blanks tests.

3.5 Procedures

Six Ph.D. students participated in a pilot test. Minor changes were made to the materials and procedures based on their feedback. During the experiment proper, participants were given a pre-questionnaire that asked their experience in UML and the domain modeled. They were also provided a summary of the UML syntax. Participants then had 10 minutes to complete a set of comprehension questions about the UML diagrams. These were not a dependent measure but merely a way of engaging participants and helping them become familiar with the diagrams before the experimental tasks. Next, participants received the fill-in-the-blanks or problem-solving test (based on random assignment). The fill-in-the-blanks test was allocated 25 minutes and the problem-solving test 35 minutes. The instructions explained that all three diagrams would be useful. When the time elapsed, participants' answers were collected and the second task was provided. After completing both tasks, participants completed the ease-of-understanding instrument. The experiment took approximately 1 hour, 15 minutes in total.

Recent research has adopted two approaches when testing diagram-based problem solving. In one approach, researchers take away participants' diagrams before the experimental tasks begin, to force participants to work from memory (e.g., Bodart et al. 2001; Gemino 1998). In a second approach, participants can access and work through the diagrams when problem solving (e.g., Kim et al. 2000). We adopted the second approach. Like Kim et al., our diagrams contained more elements in total than typically used in the first approach. Taking away the three diagrams would have made it unlikely that participants in any group could recall enough details to perform effectively. One weakness in the second approach is if participants can answer the questions by simply copying information from the diagrams, without engaging in problem-solving processes. This was alleviated in our study by requiring participants to explain each answer in the problem-solving test, and by having to integrate material from the three diagrams to complete the fill-in-the-blanks test.

4 RESULTS

The data analysis proceeded in two steps. We first examined descriptive statistics and the psychometric properties of the perceived ease-of-understanding instrument. We then performed tests of our hypotheses.

4.1 Descriptive Statistics

Table 4 reports our correlation matrix. The correlation matrix shows significant correlations ($\alpha = .05$) in the hypothesized direction for fill-in-the-blanks and problem solving. Perceived ease-of-understanding, however, showed no significant differences across groups. Participants' *experience* in UML and their perceived *knowledge* of the domain had no effect on any dependent

variable. There appeared a learning effect on the fill-in-the-blanks test, as the *order* of tasks was positive and significant.² The problem-solving results did not suffer a learning effect.

Table 4. Pearson Correlations

	Group	Order	Months Exp	Exp Bus	CLOZE	TOTPROB
Order	.000					
Months Exp	.006	-.090				
Exp Bus	.062	.084	-.199			
CLOZE	.246*	.346**	.105	-.086		
TOTPROB	.543**	.127	-.074	.018	.500**	
PEOU	.033	.157	.078	-.070	.019	.038

* Correlation is significant at the 0.05 level (1-tailed), ** significant at the .01 level (1-tailed)

Legend: Group: 1 = Bad; 2 = Moderate; 3 = Good

Order: 1 = fill-in-the-blanks first, 2 = fill-in-the-blanks second

Months Exp: How long the participant has known UML

Exp Bus: How much knowledge of the business domain the participant reports

CLOZE: Total score on the cloze or (fill-in-the-blanks test)

TOTPROB: Total number of acceptable answers in the problem-solving exercise

PEOU: Perceived ease-of-understanding

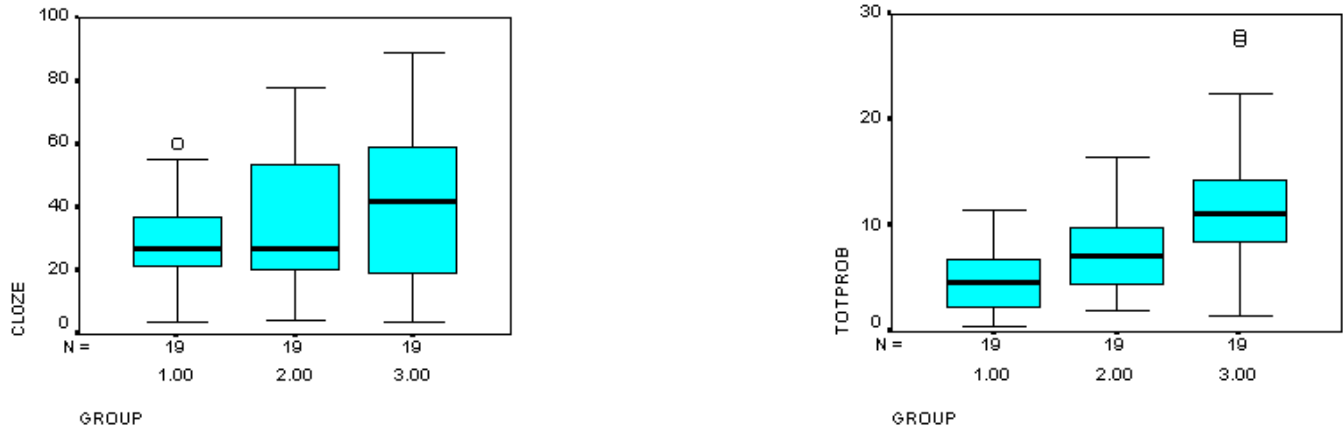
We found the perceived ease-of-understanding instrument to be unidimensional. All items in a confirmatory factor analysis loaded greater than .7. The reliability was lower than the .7 rule of thumb ($\alpha = .64$). Given the extensive validation of the ease-of-use instrument on which ours was based (Moore and Benbasat 1991), we remained somewhat confident in our scale’s psychometric properties, although we return to this issue in our conclusions.

F- and t-tests assume that dependent variables for each group have normal distribution and homogenous variance. We had three groups and three dependent variables. For two of the nine distributions, there was evidence of non-normality (significant Kolmogorov-Smirnov statistic). One of these was due to outliers, which when removed had no bearing on the results. The normality assumption when violated, however, is typically not disruptive to the functioning of t- or F-tests (Huck and Cormier 1996). Kirk (1995) reports that when cell sizes are equal, the normality assumption does not become problematic unless cell sizes drop below 12. Figure 5 shows the distribution of results for the problem-solving and fill-in-the-blanks tests. Higher quality decompositions were associated with higher variance. This does not affect the statistical tests because equal Ns make t- and F-tests robust to violations of this assumption (Huck and Cormier 1996), but we return to this finding in our conclusions.

4.2 Tests of Propositions

Table 5 reports one-way ANOVAs to test propositions 1 through 3. We also ran ANCOVAs with UML experience as a covariate (results not shown) and found the same results. Proposition 1 predicted that participants’ understanding of the diagrams would be better in the good than bad group. Our results support this proposition for the two objective measures but not for the ease-of-understanding scale. As expected, the results were less strong for the Good→Moderate and Moderate→Bad relationships; the results supported propositions 2 and 3 on problem solving but not for fill-in-the-blanks. Once again, the ease-of-understanding results showed no difference across groups.

²As suggested by a reviewer, this may have been because the narrative of the fill-in-the-blanks test activated nodes in memory that were not activated by the diagrams, thus interfering with the treatment effect. This may have been exacerbated by doing the problem-solving questions first as these would have further increased the priming of nodes in memory.



Legend: Group: 1 = Bad; 2 = Moderate; 3 = Good
 CLOZE: Total score on the cloze or (fill-in-the-blanks test)
 TOTPROB: Total number of acceptable answers in the problem-solving exercise
 Outliers: The outliers shown above did not alter the results so are included in all reported tests
 Maximum scores: the maximum scores were 116 (CLOZE) and 49 (problem-solving)

Figure 5. Descriptive Statistics for Hypothesis Tests

Table 5. One-Way ANOVAs for Differences Between Groups

Proposition	Dependent Var.	df	Mean (Standard Dev.) Between Groups	F	Sig. (1-tail)
Proposition 1: Good > Bad	CLOZE	36, 1	41.79 (25.21) (Good) > 28.71 (14.70) (Bad)	3.82	.030
	TOTPROB	36, 1	12.82 (7.08) (Good) > 5.00 (3.20) (Bad)	19.24	.000
	PEOU	35, 1	4.18 (0.73) (Good) > 4.13 (0.67) (Bad)	0.45	.417
Proposition 2: Good > Moderate	CLOZE	36, 1	41.79 (25.21) (Good) > 34.45 (23.47) (Mod.)	0.86	.180
	TOTPROB	36, 1	12.82 (7.08) (Good) > 7.34 (3.87) (Mod.)	8.75	.003
	PEOU	35, 1	4.22 (0.51) (Mod.) > 4.18 (0.73) (Good)	0.44	.418
Proposition 3: Moderate > Bad	CLOZE	36, 1	34.45 (23.47) (Mod.) > 28.71 (14.70) (Bad)	0.82	.187
	TOTPROB	36, 1	7.34 (3.87) (Mod.) > 5.00 (3.20) (Bad)	4.14	.025
	PEOU	36, 1	4.22 (0.51) (Mod.) > 4.13 (0.67) (Bad)	0.23	.319

Legend: CLOZE: Total score on the cloze or (fill-in-the-blanks test)
 TOTPROB: Total number of acceptable answers in the problem-solving exercise
 PEOU: Perceived ease-of-understanding

Table 6 reports the results for our fourth proposition: that decomposition quality and understanding are positively linearly related (i.e., can be fitted by a regression line). The results support this proposition for the fill-in-the-blanks and problem-solving measures as both regression models are significant, as are the coefficients. The results also confirmed that the problem-solving measure was more powerful than the fill-in-the-blanks test. The adjusted R² for fill-in-the-blanks (.15) was half that for problem-solving (.28), even taking into account *order* (learning) effects.

As shown in Figure 5, participants generally received low scores in the tasks, consistent with the challenging nature of the experiment. To provide further confidence in the results, we ran two sensitivity tests (not shown here to conserve space). First, we tested the consistency of the problem-solving results across the 11 questions. The results appeared consistent as we found the score for every problem-solving question was positively correlated with the quality of the decomposition, and in a MANOVA

with all 11 questions, eight were significant and in the hypothesized direction ($p < .05$, one-tailed). Second, we examined whether the results differed depending on the diagram used to answer the question. Recall that there were three types of questions based on the number of diagrams used to create their acceptable answers (refer to section 3.4). Of the three types of questions, both questions with answers from one diagram received support, five of the six questions with answers from two diagrams received support, and one of the three questions with answers from all three diagrams received supported. It, therefore, did not appear that the results were skewed by the use of only one particular diagram.

Table 6. Regression Results (for Proposition 4)

Cloze (Fill-in-the-Blanks):	Coefficient	Std. Error	t	Sig. (1-tail)	
	(Constant)	-.762	10.738	-.071	.472
	GROUP	6.539	3.277	1.996	.026
	ORDER	15.023	5.352	2.807	.004
Predictors: (Constant), ORDER, GROUP (1 = Bad, 2 = Moderate, 3 = Good)					
Overall Model: $F = 5.931$ ($p = .003$) ($df = 54, 2$), $Adj R^2 = .150$					
When ORDER excluded, GROUP remains significant but the $Adj R^2$ drops to .043					
Problem-solving	Coefficient	Std. Error	t	Sig. (1-tail)	
	(Constant)	.570	1.759	.324	.374
	GROUP	3.908	.814	4.799	.000
	ORDER				
Predictors: (Constant), GROUP (1 = Bad, 2 = Moderate, 3 = Good)					
Overall Model: $F = 23.034$ ($p < .001$) ($df = 55, 1$), $Adj R^2 = .282$					
ORDER was not included in regression because it was not significant					

Table 7 summarizes our results. We received support for each of proposition subject to two trends: (1) the problem-solving measure was the most powerful measure of understanding, yielding consistent support for all propositions; and (2) no significant differences were found for perceived ease-of-understanding in any of our tests.

Table 7. Summary of Results:

Dependent Variable	Proposition 1: Good > Bad	Proposition 2: Good > Moderate	Proposition 3: Moderate > Bad	Proposition 4: Better decompositions increase understanding?
Problem-solving	Yes	Yes	Yes	Yes
Fill-in-the-blanks	Yes	No	No	Yes
Ease-of-Understanding	No	No	No	No

5 CONCLUSION AND IMPLICATIONS

The implications of this study should be considered in light of its limitations. First, like all laboratory experiments, generalizations should be made cautiously. We note that our participants were less experienced and likely less motivated than practicing analysts. Second, the results are for only one experiment and a small sample. Replications would be desirable. Third, we tested each violation of the good decomposition conditions in limited ways. A broader test could test each violation in more depth. Fourth, our participants worked under tight time constraints. Further research is needed to determine the degree to which the theory holds as time constraints are reduced and participants have more expertise. Finally, our diagrams combined several violations in the one model. A recent study of program understanding found weak effects when testing each violation on its own (Weber 2001). While the difference in results could be due to different tasks or procedures, it may also be due to stronger effects occurring when there is an interaction of violations. Our experimental design did not allow us to test these effects (we acknowledge an anonymous reviewer for this insight).

Despite these limitations, we believe our results bear implications for practitioners and researchers. We consider the main benefit of the good decomposition model for practitioners to be its parsimony and wide applicability. As UML includes a massive collection of techniques (Kobryn 1999; Wieringa 1998), a parsimonious metric can help practitioners focus on how each technique adds or detracts from an overall system decomposition. We believe our initial results are strong enough for practitioners to consider testing the good decomposition model as an OOA quality metric. The conditions could also be used as a training device to help analysts develop good UML diagrams.

For researchers, our study offers four avenues of research. First, our results for the perceived ease-of-understanding instrument showed no difference in results across the groups, consistent with Gemino (1998) and Burton-Jones and Weber (1999). This raises two possibilities: (1) the instrument may lack reliability or tap into the wrong construct to test understanding, or (2) the relationship between *actual* and *perceived* understanding is more complex than initially assumed. Studying these implications and, potentially, designing an improved instrument, would be very useful for future research. Second, on two dependent variables, we found the decomposition conditions supported understanding but also *increased* variance; the benefit of good decompositions was not the same for all participants (refer to Figure 5). Future research could test whether this was an artifact of our design or whether individual differences might be interacting with decomposition quality to produce the observed effects. Third, it is known that the use of conceptual models in practice differs greatly from academic prescriptions. Redesigning the study for a practitioner environment would be most valuable. Fourth, we note Bansiya and Davis' (2002) recent objection that many quality metrics for object-oriented software analyses can be applied only after a product is complete, or nearly complete. They called for metrics to support the early stages of development to ensure that the analysis and design had favorable internal properties. Our results provide some evidence that the good decomposition conditions could form such a metric for OOA. Future research will hopefully test this hypothesis further.

6 ACKNOWLEDGEMENTS

We are indebted to Ron Weber, Yair Wand, and Bala Ramesh for advice and encouragement during our research. We thank the SE, the AE, and the three anonymous reviewers for many excellent comments. The usual disclaimer applies.

7 REFERENCES

- Agarwal, R., De, P., Sinha, A. P., and Tanniru, M. "On the Usability of OO Representations," *Communications of the ACM* (43:10), 2000, pp. 83.
- Ashcraft, M. H. *Cognition*. Upper Saddle River, NJ: Prentice-Hall, 2002.
- Bansiya, J., and Davis, C. G. "A Hierarchical Model of Object-Oriented Design Quality Assessment," *IEEE Transactions on Software Engineering* (28:1), 2002, pp. 4-17.
- Basili, V., Briand, L. C., and Melo, W. L. "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Transactions on Software Engineering* (22:10), 1996, pp. 751-761.
- Bodart, F., Patel, A., Sim, M., and Weber, R. "Should Optional Properties Be Used in Conceptual Modeling? A Theory and Three Empirical Tests," *Information Systems Research* (12:4), 2001, pp. 384-405.
- Bunge, M. *Treatise on Basic Philosophy: Volume 3: Ontology 1: The Furniture of the World*. Boston: Reidel, 1977.

- Burton-Jones, A., and Weber, R. "Understanding Relationships with Attributes in Entity-Relationship Diagrams," in P. De and J. I. DeGross (eds.), *Proceedings of the Twentieth International Conference on Information Systems*, Charlotte, NC, 1999, pp. 214-228.
- Chidamber, S. R., and Kemerer, C. F. "A Metrics Suite for Object-Oriented Design," *IEEE Transactions on Software Engineering* (20:6), 1994, pp. 476-493.
- Chuang, T.-T., and Yadav, S. B. "A Decision-Driven Approach to Object-Oriented Analysis," *The DATA BASE for Advances in Information Systems* (31:2), 2000, pp. 13-34.
- Collins, A. M., and Quillan, M. R. "Retrieval Time from Semantic Memory," *Journal of Verbal Learning and Behavior* (8), 1969, pp. 240-247.
- Conger, S. A. *The New Software Engineering*. Belmont, CA: Wadsworth, 1994.
- Davis, F. "Perceived Usefulness, Perceived Ease of Use, and End User Acceptance of Information Technology," *MIS Quarterly* (13:3), 1989, pp. 318-339.
- Dromney, R. G. "Cornering the Chimera," *IEEE Software* (13:1), 1996, pp. 33-43.
- Gemino, A. "To Be or May to Be: An Empirical Comparison of Mandatory and Optional Properties in Conceptual Modeling," *Proceedings of the Annual Conference of the Administrative Sciences Association of Canada, Information Systems Division* (19:4), Saskatoon, Saskatchewan, May 30–June 2, 1998, pp. 33-44.
- Hardgrave, B., and Dalal, N. "Comparing Object-Oriented and Extended-Entity-Relational Data Models," *Journal of Database Management* (6:3), 1995, pp. 15-21.
- Huck, S. W., and Cormier, W. H. *Reading Statistics and Research*. New York: Harper Collins, 1996.
- Johnson, R. A. "Object-Oriented Systems Development: A Review of Empirical Research," *Communications of the AIS* (8), 2002, pp. 65-81.
- Kim, J., Hahn, J., and Hahn, H. "How do we Understand a System with (So) Many Diagrams? Cognitive Integration Processes in Diagrammatic Reasoning," *Information Systems Research* (11:3), 2000, pp. 284-303.
- Kirk, R. E. *Experimental Design: Procedures for the Behavioral Sciences* (3rd ed.). Pacific Grove, CA: Brooks/Cole Publishing Co., 1995.
- Kobryn, C. "UML 2001: A Standardization Odyssey," *Communications of the ACM* (42:10), 1999, pp. 29-37.
- Mayer, R. "Models for Understanding," *Review of Educational Research* (59), 1989, pp. 43-64.
- Moore, G. C., and Benbasat, I. "Development of an Instrument to Measure the Perceptions of Adopting and Information Technology Innovation," *Information Systems Research* (2), 1991, pp. 192-222.
- Parnas, D. L. "On the Criteria to be Used in Decomposing Systems into Modules," *Communications of the ACM* (15:12), 1972, pp. 1053-1058.
- Parsons, J., and Wand, Y. "Using Objects for Systems Analysis," *Communications of the ACM* (40:12), 1997, pp. 104-110.
- Paulson, D., and Wand, Y. "An Automated Approach to Information Systems Decomposition," *IEEE Transactions on Software Engineering* (18:3), 1992, pp. 174-189.
- Shanks, G. "Representation in Information Systems," paper presented to the IS Foundations Workshop, Department of Computing, Macquarie University, 1999.
- Siau, K., and Tian, Y. "The Complexity of Unified Modeling Language: A GOMS Analysis," in V. Storey, S. Sarkar, and J. I. DeGross (eds.), *Proceedings of the Twenty-second International Conference on Information Systems*, New Orleans, LA, 2001, pp. 443-447.
- Sircar, S., Nerur, S. P., and Mahapatra, R. "Revolution or Evolution? A Comparison of Object-Oriented and Structured Systems Development Methods," *MIS Quarterly* (25:4), 2001, pp. 457-471.
- Vessey, I., and Conger, S. A. "Requirements Specification: Learning Object, Process, and Data Methodologies," *Communications of the ACM* (37:5), 1994, pp. 102-113.
- Wand, Y., and Weber, R. "A Model of Systems Decomposition," in J. I. DeGross, J. C. Henderson, and B. R. Konsynski (eds.), *Proceedings of the Tenth International Conference on Information Systems*, Boston, MA, 1989, pp. 42-51.
- Wand, Y., and Weber, R. "Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda," *Information Systems Research* (13:4), 2002.
- Weber, R. "Comprehending Decompositions: A Theory and Two Empirical Tests," Working Paper, University of Queensland, 2001.
- Weber, R. *Ontological Foundations of Information Systems*. Melbourne: Coopers & Lybrand and Accounting Association of Australia and New Zealand, 1997.
- Wieringa, R. "A Survey of Structured and Object-Oriented Software Specification Methods and Techniques," *ACM Computing Surveys* (30:4), 1998, pp. 459-527.