Scandinavian Journal of Information Systems

Volume 23 | Issue 1 Article 3

6-30-2011

Complementary Use of Modeling Grammars

Peter Green
The University of Queensland, p.green@business.uq.edu.au

Michael Rosemann
Queensland University of Technology, m.rosemann@qut.edu.au

Marta Indulska
University of Queensland, m.indulska@business.uq.edu.au

Jan Recker
Queensland University of Technology, j.recker@qut.edu.au

Recommended Citation

Green, Peter; Rosemann, Michael; Indulska, Marta; and Recker, Jan (2011) "Complementary Use of Modeling Grammars," *Scandinavian Journal of Information Systems*: Vol. 23 : Iss. 1 , Article 3.
Available at: http://aisel.aisnet.org/sjis/vol23/iss1/3

Complementary Use of Modeling Grammars

Peter Green
UQ Business School, The University of Queensland, Australia
p.green@business.uq.edu.au

Michael Rosemann
Information Systems Discipline, Queensland University of Technology, Australia
m.rosemann@qut.edu.au

Marta Indulska
UQ Business School, The University of Queensland, Australia
m.indulska@business.uq.edu.au

Jan Recker
Information Systems Discipline, Queensland University of Technology, Australia
j.recker@qut.edu.au

Abstract: Conceptual modeling continues to be an important means for graphically capturing the requirements of an information system. Observations of modeling practice suggest that modelers often use multiple modeling grammars in combination to articulate various aspects of real-world domains. We extend an ontological theory of representation to suggest why and how users employ multiple conceptual modeling grammars in combination. We provide an empirical test of the extended theory using survey data and structured interviews about the use of traditional and structured analysis grammars within an automated tool environment. We find that users of the analyzed tool combine grammars to overcome the ontological incompleteness that exists in each grammar. Users further selected their starting grammar from a predicted subset of grammars only. The qualitative data provides insights as to why some of the predicted deficiencies manifest in practice differently than predicted.

Keywords: ontological theory of expressiveness; minimal ontological overlap; maximal ontological completeness, conceptual modeling.

Accepting editor: Bjørn Erik Munkvold

1 Introduction

A major task undertaken by systems analysts/designers, workflow engineers, process modelers, and the like, is to develop a conceptual model of a perception(s) of a portion of the world to analyze, design or configure an information system intended to simulate or support the real-world area under investigation. The models are specified using "a grammar (i.e., a set of constructs and rules to combine those constructs) and a method (i.e., procedures by which the grammar can be used)" (Wand and Weber 2002).

As the importance of modeling for information systems (IS) analysis and design has been recognized over time, researchers have increasingly attempted to provide some guidance and insight to assist practitioners in the comparison, evaluation, and use of such grammars (Moody 2005). In this vein of research, scholars have increasingly drawn upon theoretical work by Wand and Weber (1990; 1993) on a theory of ontological expressiveness to design, compare and evaluate modeling grammars (e.g., Bowen et al. 2009; Shanks et al. 2008).

The studies to date have in common that they examined how a *single* modeling grammar allows users to create complete and/or clear conceptual models of real-world domains. Studies of conceptual modeling in practice (Fettke 2009), however, frequently report that modelers often use multiple grammars in combination. This situation is because users have a need to model various portions of a real-world domain from different perspectives (e.g., the behavior of important agents, the structure of important real-world concepts, or their associations). To that end, they use a variety of grammars to create different models of real-world systems.

In this paper, therefore, we extend Wand and Weber's theory of ontological expressiveness (e.g., Weber 1997) to provide an explanatory model to describe the use of multiple conceptual modeling grammars in combination. Specifically, we utilize an extension, known as *overlap analysis,* to describe how and why modelers use multiple grammars when developing graphical descriptions of real-world domains intended to be supported by an information system. We then examine empirically our theoretical predictions using data gathered about the use of multiple grammars in an automated tool environment. Automated tool environment are important to modeling because there is evidence to suggest that these tools continue to be used and have a positive impact on systems development effectiveness at those sites where they are used (Chau 1996; Premkumar and Potter 1995). But while research has established a body of knowledge around the use of modeling tools (Iivari 1996; Sørensen 1993), or the relationships between modeling tools and systems development methodologies (Vessey et al. 1992), and while IS research has examined the use of single grammars for modeling as part of systems development (Green and Rosemann 2004), no research to date has examined the use of multiple grammars in an automated tool environment, which is the interest in our study.

Specifically, we identify the following two research questions:

1. Why do modelers use multiple conceptual modeling grammars in combination when modeling for systems development?
2. Which conceptual modeling grammars are used in combination by modelers when modeling for systems development?

This paper unfolds in the following manner. Section 2 provides an introduction to the selected theoretical foundation of our study and reviews related work on the basis of the selected theory. The next section explains the method of ontological analysis associated with the theory and, in particular, the method extension known as overlap analysis. We then discuss the development of our research hypotheses. Section 4 describes the research method used, and Section 5 reports on the results and discusses the findings. In Section 6 we summarize the main contributions, review potential limitations of the work and outline implications of the study for research and practice.

2 Background

2.1 Theoretical foundations

Systems analysts/designers often use different grammars for modeling different aspects of a real-world domain intended to be supported by an information system. For instance, they may choose to model the data structure using an Entity-Relationship diagram, and a Business Process Diagram to model relevant organizational processes in the domain. While this observation seems obvious, to date there is no theoretical model available to explain and predict why and how modelers choose different grammars to model different aspects of real-world domains.

We turn to a theory of ontological expressiveness (Wand and Weber 1993) in order to facilitate such an explanation. This theory was developed from the adaptation of an ontology proposed by Bunge (1977). With information systems essentially being human-created representations of real-world systems, Wand and Weber (1990) suggest that ontology may help in devising conceptual structures on which modelers can base their representations of these systems.

Following the assumption of a representational nature of an information system, Wand and Weber (1990) suggest an ontological model of representation, known as the Bunge-Wand-Weber (BWW) representation model, which specifies a set of rigorously defined ontological constructs to describe all types of real-world phenomena that a modeling grammar user may desire to have articulated in a conceptual model. A description of Wand and Weber's model is provided in (Weber 1997).

Based on their ontological model, Wand and Weber (1993) developed a theory of ontological expressiveness that suggests that any modeling grammar must be able to represent all things in the real world that might be of interest to users of information systems; otherwise, the resultant model is incomplete, and the analyst/designer will somehow have to augment the model(s) (Weber 1997). Wand and Weber (1993) clarify two major situations that may occur: lack of ontological completeness (i.e., construct deficit) and lack of ontological clarity (i.e., construct overload, construct redundancy, and construct excess).

Several researchers have empirically tested this argument. Recker et al. (2010), for instance, found that construct deficit motivated grammar users to employ additional means to help articulate the real world phenomena they felt could not be expressed with the grammar in use.

Bodart et al. (2001) showed how the existence of construct excess in a conceptual model results in users making more understanding errors when interpreting the model. Similarly, Shanks et al. (2008) demonstrated that construct overload undermines users' ability to understand the information contained in the model.

In this paper, our interest specifically lies in the concept of construct deficit, and how construct deficit can assist an understanding why and how modelers use multiple grammars provided within a modeling tool.

A grammar exhibits construct deficit unless there is at least one grammatical construct for each construct in the ontological model of representation. The main premise associated with construct deficit is that grammar users will tend to employ additional means of articulation in order to compensate for the deficit (e.g., via additional grammars, textual descriptions or other means) (Weber 1997). We would therefore suggest that modelers use multiple grammars in combination to offset potential ontological incompleteness found in any single grammar.

Note that Wand and Weber's (1993) theory of ontological clarity is not a neurophysiological theory that explains the cognitive processes modelers engage in when making a decision about which grammars to use for modeling (Shanks et al. 2008). Yet, their theory provides a potential rationale for *why* users make such grammar usage choices – because the grammars they work with have certain deficiencies, which, in turn, motivates users to make decisions about the use of additional modelling means.

2.2 Related work

Wand and Weber's (1993) theory has been used in various modeling domains e.g., traditional, structured, data-oriented, object-oriented, process modeling, activity-based costing, ERP systems, Enterprise systems interoperability, other ontologies, Use cases, and Reference models (Recker 2011). Most of the work to date involved the analysis of a single grammar. Also, much of the work has been analytical in nature, with few of the studies validating their results through qualitative and/or quantitative empirical tests. Notable exceptions are those in (Bowen et al. 2009; Burton-Jones and Meso 2006; Recker et al. 2011; Shanks et al. 2008).

No work to date has considered Wand and Weber's (1993) theory in empirical studies of multiple grammars. To address this gap, we will use and extend two important concepts relevant to Wand and Weber's (1993) theory—*minimal ontological overlap* (MOO) and *maximal ontological completeness* (MOC). MOC and MOO were first described by Green (1997) but were not fully operationalised and used in an analytical sense until Green et al. (2007) used them to explain how various interoperability standards might best be used in combination to overcome representational deficiencies inherent in the individual standards considered. The work presented in this paper now extends this work in that it enhances that operationalisation of MOO and MOC and provides the first test of these theoretical principles by examining data collected on the use of traditional and structured grammars in combination within an automated tool environment.

3 Analysis and hypotheses

3.1 Methodology

To derive hypotheses to suggest why and how modelers user use multiple grammars within an automated tool environment, we first have to establish to what extent the grammars available share representations for important ontological constructs. This process is known as an *ontological analysis* (Wand and Weber 1993), which we extend in this paper to also consider the extent, and type, of ontological overlap.

Our analysis methodology, specifically, consists of three steps.

First, we perform a traditional ontological analysis of all grammars available in the modeling tool environment we consider. The objective is to identify those grammar constructs, in each grammar, that have mappings to Wand and Weber's ontological model that are not isomorphic. These situations indicate the presence of construct deficit. Construct deficit, in turn, suggests that the grammar is ontologically incomplete, meaning that there is a portion of the real-world that users will not be able to articulate in a conceptual model.

Second, we apply the process of overlap analysis (Green et al. 2007) in order to determine the combinations of modeling grammars that provide, as per theory, the lowest levels of ontological incompleteness between them.

Overlap analysis allows scholars to identify two important characteristics of modeling grammar combinations. The first, maximum ontological completeness (MOC), states that two grammars afford together the maximum ontological expressiveness, when they, together, have the lowest level of construct deficit amongst all possible combinations of modeling grammars. The second, minimal ontological overlap (MOO), states that, when selecting combinations of grammars that achieve maximum ontological completeness, those combinations are superior that have a minimal overlap in the representation of ontological constructs, i.e., grammar combinations where few, if any, grammatical constructs from both grammars map to the same ontological construct.

Accordingly, in our second step, we examine the results from the ontological analysis to identify those combinations of grammars that: (a) achieve maximum ontological completeness, and (b) afford minimal ontological overlap.

Third, we then examine the results from the overlap analysis to develop specific hypotheses about why and how modelers use multiple grammars in combination, as per the predictions offered by the theory.

3.2 Ontological analysis

We consider the grammars implemented in a popular automated analysis and design tool with a well-established user base in Australasia—Excelerator (v1.9) from Intersolv Inc. We selected Excelerator as a representative for structured upper automated modeling tools for three reasons. First, Excelerator was the first automated modeling tool available for personal computers in the early nineties, and has since then been refined to contain the latest versions of modeling grammars (Ricciuti 1992). Second, over the years in which the data collection was conducted, Excelerator has traditionally been a top selling automated modeling tool in Australia (Martin and De Luca 1992), therefore increasing confidence that a large user base would be available for this study. Third, Excelerator is an example of a methodology companion (Vessey et al. 1992), providing support for maintaining consistency across multiple versions of a design, similar to other market-leading tools (e.g., Texas Instruments).

Nine traditional and structured graphical grammars in this tool were identified, *viz.*, system flowchart (SF), program flowchart (PF), logical data flow diagram (LDFD), structure chart (STC), state transition diagram (TRD), structure diagram (STD), structured decision table (SDT), entity-relationship diagram (ERA), and data model diagram (DMD). Our objective was thus to perform an ontological analysis of these nine grammars. Specifically, we proceed in two steps.

First, for each grammar, we performed a representation mapping (Wand and Weber 1993)—a mapping of the ontological constructs specified in Wand and Weber's model of representation to constructs contained in each of the modeling grammars—in order to identify those mappings that are not isomorphic. Wherever we found that an ontological construct in Wand and Weber's model did not have a related representation construct in a grammar, we indicated this situation as *construct deficit* in the grammar. Situations in which one ontological construct was represented by multiple grammatical constructs denote situations of *construct redundancy*.

Second, we performed an interpretation mapping (Wand and Weber 1993)—a mapping of the remaining grammatical constructs in each of the modeling grammars to ontological constructs. This mapping can lead to situations where one grammatical construct corresponds to several ontological constructs (construct overload) or where for one grammatical construct no corresponding ontological construct can be identified (construct excess). The mappings were performed by one of the authors. We note that the reliability of the mappings could have been strengthened by using a multi-coder mapping procedure as described in (Recker et al. 2010).

We illustrate our mapping procedure by considering the application of the analysis using the case of the Entity-Relationship Diagram (ERA) grammar. The same reasoning applies almost exactly to the mappings for the constructs in the Data Model Diagram (DMD), except for the relationship type, which is binary for DMD but n-ary for ERA. The complete discussion of the conduct of the analysis for all grammars is omitted for the sake of brevity but is available from the authors on request.

The mapping process was performed by careful reading the ERA grammar specification as well as Wand and Weber's ontological construct specifications. Based on this understanding, mappings were then identified in cases where a grammatical construct was perceived to correspond to the definition of one or more ontological constructs. For instance, in the ERA grammar, based on the grammar and ontological construct definitions, we believe that the ontological construct class is represented by a data entity type. This is because, for example, a CUSTOMER entity type on an ERA represents customers that share the common, single property of being customers of the company of interest. A type of state law is represented by the cardinality constraints on a data relationship. It constrains the values of the *binding mutual property* (or *coupling*) of the things by specifying how many 'replications' of this property each of the coupled things must (or can) have. Optionality constraints only exist in ERA diagrams because ERA diagrams do not represent individual things. Rather, the data entity type represents *classes* of things. Optionality simply says that some individual things in a coupled class may or may not participate in the coupling. A coupling (or binding mutual property) is represented by a data n-ary relationship. While there are no specific constructs for thing or property of a thing in ERA diagrams, the data dictionary augments the grammar by providing integrated record definitions (through the REC construct) and data item definitions (through the ELE construct). The REC construct is interpreted as representing a *thing*, while the ELE construct is interpreted as a *property*. If a data element describes the *interaction* (or coupling) of two or more entities, Yourdon (1989) prescribes that the 'naked' relationship between the entities should be replaced with an *associative entity type*. An associative entity type can be assigned attributes (data elements) of its own, and can participate in further relationships. The associative entity type is an artificial mechanism by which n-ary (n > 2) relationships in the real world are represented in the model as a series of binary relationships. It represents a number of binding mutual properties (or couplings).

Using similar reasoning, we completed the analysis of representation and interpretation mappings for all grammars implemented in Excelerator (see Table 1 for summary). Interpretation mappings of grammatical constructs to ontological constructs are denoted with an "(I)".

To prepare step two of our methodology, the overlap analysis, we then considered the mapping results with a specific interest in all occurrences of construct deficit. Table 2 provides an overview of the analysis results and highlights those constructs that do not have a grammatical representation in any of the nine grammars, *viz.*, conceivable state space, lawful state space, conceivable event space, lawful event space, history, unstable state, and poorly-defined event.

In light of Wand and Weber's (1993) theory, not having representations for these ontological real-world constructs at the time of modeling the information systems solution will cause problems for the system at later stages of the development life cycle. For example, the rules that define the lawful state space, and consequently, the lawful event space of a thing are important in the design of an information system. These rules are referred to in the practice of systems analysis and design as business rules (e.g., von Halle 2001). Still, the identification, recording, and integration of the relevant business rules into the design of an information system remain poorly-handled issues that manifest themselves in systems poorly received by end-users. Our contention therefore is that users will seek additional modeling means to express those real-world phenomena they feel cannot be expressed with an available grammar because of a deficit of representation constructs. This reason would motivate a user to use a combination of grammars that provides the representation capability missing in their first chosen grammar, or to extend the grammars with additional constructs if a combination of grammars is still unable to provide the required representation (e.g., representation for lawful state space in this case). The former situation is what we consider in the following.

3.3 Overlap analysis

Having identified the extent of ontological completeness in each of the grammars, our next step is to consider the type and extent of ontological overlap in combinations of grammars available in Excelerator.

We proceeded as follows:

Ontological Construct	*SF*	*PF*	*LDFD*	*STC*	*TRD*	*STD*	*SDT*	*ERA*	*DMD*
	System flowchart	Program flowchart	Data flow diagram	Structure chart	State transition diagram	Structure diagram	Structured decision diagram	Entity-relationship diagram	Data model diagram
*Thing**	No construct in SF, described by REC in XLDict(I)	No construct in PF, described by REC in XLDict(I)	No construct in LDFD, described by REC in XLDict(I)	No construct in STC, described by REC in XLDict(I)		No construct in STD, described by REC in XLDict(I)		No construct in ERA, described by REC in XLDict(I)	No construct in DMD, described by REC in XLDict(I)
Property:* *Intrinsic* *Non-binding* *Mutual* *Emergent* *Hereditary* *Attributes*	No construct in SF, described by ELE in XLDict(I)	No construct in PF, described by ELE in XLDict(I)	No construct in LDFD, described by ELE in XLDict(I)	1. Data couple 2. ELE in XLDict(I)		No construct in STD, described by ELE in XLDict(I)		No construct in ERA, described by ELE in XLDict(I)	No construct in DMD, described by ELE in XLDict(I)
Class	1. Media objects(I) 2. People objects		1. Data store 2. Data flow 3. Ext. entity	1. System device(I) 2. Global data store				Entity type	Entity
*State**		Flowline	1. Data flow(I) 2. Control flow(I)	Data couple(I)	State	Connection (I)			
State law: *Stab. condition* *Corr. action*		Condition Process; I-O process		1. Decision diamond 2. Control couple Function		Select Function; Inclusive alternative		Cardinality of relations HIP	1. Cardinality 2. Binary relations HIPs

Ontological Construct	*SF*	*PF*	*LDFD*	*STC*	*TRD*	*STD*	*SDT*	*ERA*	*DMD*
Event		In flowline -> (Process/I-O process) -> Out flowline	1. At funct. prim. level, In data flow -Process - Out data flow 2. Ext. entity -> Data flow 3. Ext. entity -> Ctl flow (signal) 4. Ctl flow (signal) - Ctl transform - Ctl flow (Prompt)	At lowest level, In data couple - function - out data couple	1. State transition vector - state 2. “Naked” transition vector 3. Condition	1. At lowest level, In connection - function - connection (I) 2. Release condition 3. Terminating activity			
*Transformation**		Process rectangle	At func. prim. level, Process	Function	Action	At the lowest level, Function, Inclusive alternative, parallel activity	Action		
Coupling: Binding mutual property								1. N-ary relations HIP type 2. Associative entity(I) type	1. Binary rel’ship
*Stable state**		Flowline (I)	1. Data flow(I) 2. Ctl flow(I)	Data couple(I)	States on TRD	Connection (I)			

Table 1. Portion of the grammar analysis (Note: * indicates a fundamental ontological construct (Wand and Weber 1995))

Ontological construct	*SF*	*PF*	*LDFD*	*STC*	*TRD*	*STD*	*SDT*	*ERA*	*DMD*
Thing	✓	✓	✓	✓		✓		✓	✓
Property	✓	✓	✓	✓		✓		✓	✓
Class	✓		✓	✓				✓	✓
Kind								✓	✓
State		✓	✓	✓	✓	✓			
Conc. state space									
State law - mono-property		✓		✓		✓			
Lawful state space									
Event		✓	✓	✓	✓	✓			
Process	✓	✓	✓	✓	✓	✓			
Conc. event space									
Transformation		✓	✓	✓	✓	✓	✓		
Lawful transformation		✓		✓	✓	✓	✓		
Lawful event space									
History									
Acts on								✓	✓
Coupling								✓	✓
System								✓	✓
System composition	✓		✓	✓				✓	✓
System environment			✓	✓				✓	✓
System structure								✓	✓
Subsystem	✓	✓	✓	✓	✓	✓		✓	✓
System decomposition	✓	✓	✓	✓	✓	✓	✓	✓	✓
Level structure	✓	✓	✓	✓	✓	✓	✓	✓	✓
External event	✓	✓	✓	✓	✓	✓	✓		
Stable state		✓	✓	✓	✓	✓			
Unstable state									
Internal event		✓	✓	✓	✓	✓			
Well-defined event		✓	✓	✓	✓	✓			
Poorly-defined event									
Number of ontological constructs represented	9	15	16	18	12	5	15	13	13

Table 2. Grammar analysis summary

1. We identify those combinations of grammars that are as ontologically complete as the set of available grammars will allow—a *maximally ontologically complete* (MOC) combination. As Table 2 shows, there are some ontological constructs for which none of the nine available grammars in the automated tool have a representation. Accordingly, any combination of grammars from within the tool can only be as ontologically complete as possible, rather than fully ontologically complete (of course, depending on the grammars and the tool environment, maximal completeness may be the same as full ontological completeness). In the case of Excelerator, we note from Table 2 that the combination of LDFD and STD provides maximum ontological completeness (34 ontological constructs have representations across the two grammars), and that combinations of PF with STC, and STC with SDT, provide combined representations for 33 ontological constructs.
2. MOC may be achieved through combinations of two, three, four, up to the total number of grammars in the tool. This situation implies that some of the MOC combinations include a larger set of grammatical constructs than others. Therefore, we use the rule of parsimony to select the combination of grammars with the least number of different grammars achieving maximal ontological completeness.
3. Last, we consider for the MOC combinations the extent of construct overlap between the grammars, i.e., situations in which grammatical constructs are available in both grammars to represent one particular ontological construct. *Minimal ontological overlap* would exist when the occurrence of such construct overlap can be minimized. Table 3 displays the results from the minimal ontological overlap analysis, showing that the combination of LDFD and STC has maximum ontological overlap (16 ontological constructs overlap between the two grammars), and the combination of SDT with either ERA or DMD provides minimal ontological overlap (0 overlapping ontological constructs).
4. Last, the results from both MOC and MOO analysis on basis of the results summarized in Table 2 and Table 3 are inspected to identify those combinations of grammars that provide maximum ontological completeness and minimum ontological overlap. Table 2 is inspected to determine, for each grammar, the other grammar required to maximize ontological completeness. At the same time, the results are cross-referenced to Table 3 to ensure the potential grammar combination has minimal ontological overlap. The resulting grammar combination providing both MOC and MOO are displayed in Table 4.

For example, if an analyst begins the modeling task with ERA, then STD must be combined to represent *states*, *state laws*, *events*, *processes*, *transformations*, and *lawful transformations*, thereby achieving MOC. At the same time, Table 3 shows that ERA and STD have an ontological overlap of only three constructs. The only other grammar with less overlap than STD is SDT or TRD, which, however, in combination with ERA will not produce a maximum ontologically complete set of representations.

Grammar combination	*SF*	*PF*	*LDFD*	*STC*	*TRD*	*STD*	*SDT*	*ERA*	*DMD*
SF	0								
PF	7	0							
LDFD	9	13	0						
STC	9	15	16	0					
TRD	5	12	11	12	0				
STD	7	15	13	15	12	0			
SDT	1	3	2	3	3	3	0		
ERA	7	4	8	7	3	4	0	0	
DMD	7	4	8	7	3	4	0	14	0

Table 3. Minimal ontological overlap analysis summary

Grammar combination	*SF*	*PF*	*LDFD*	*STC*	*TRD*	*STD*	*SDT*	*ERA*	*DMD*
SF		✓				✓		✓	✓
PF									
LDFD		✓				✓		✓	✓
STC								✓	✓
TRD									
STD									
SDT									
ERA		✓				✓			
DMD		✓				✓			

Table 4. Potential combinations providing MOC and MOO

3.4 Hypotheses

We argue that the principle of minimal ontological overlap can provide theoretical rationale for why, and in which combination, modelers use multiple grammars. Our main contention is that modelers use multiple grammars in combination because they seek to achieve maximal ontological completeness. In doing so, our contention is that they will seek to minimize potential ontological overlap so as to maintain efficiency in their modeling. In the following, we detail this conjecture in a number of research hypotheses.

In considering why users employ multiple grammars in combination, Table 2 demonstrates that each of the nine grammars in the automated tool is ontologically incomplete, i.e., each grammar alone displays construct deficit. Wand and Weber's (1993) theory suggests that modelers will, consciously or not, overcome the construct deficit of their first-chosen grammar by combining it with other grammars to represent the required ontological constructs. Therefore, our main contention as to *why* conceptual modelers use multiple grammars implemented in an automated tool environment is that they do so in order to achieve maximal ontological completeness, i.e., to mitigate, wherever possible, construct deficit recognized in each of the single grammars available in the automated tool. Therefore,

H1 Modelers will use two or more grammars in combination when completing their modeling tasks within Excelerator, because they experience manifestations of ontological incompleteness in the grammars used in isolation.

In considering which grammars are used in combination by modelers in their modeling efforts when having multiple grammars available, Wand and Weber (1993) emphasize the fundamental importance of the ontological constructs *system*, *system composition*, *system environment*, and *system structure* when modeling domains intended to be supported by information systems. These ontological constructs allow users to specify representationally the system that is being analyzed/designed, its structure, how it is composed, and how it is placed within its environment. When modeling information systems, identifying the type, structure, composition and environment of the system to be analyzed/designed is the most fundamental aspect of the analysis/design work (Burton-Jones and Meso 2006). Therefore, we argue that, when engaging in a modeling project with multiple grammars at hand modelers will begin their modeling efforts with a grammar that facilitates the modeling of the characteristics of the system (e.g., its structure, its composition or its environment). Therefore, they will start their modeling using a grammar that provides representations for a system-related ontological construct. Inspection of Table 2 shows that the grammars *system flowchart*, *logical data flow diagram*, *structure chart*, *entity-relationship diagram*, and *data model diagram* all provide representations for these system-related ontological constructs. We thus advance the following hypothesis:

H2 A modeler will begin the modeling effort with one of the following grammars: system flowchart, logical data flow diagram, structure chart, entity-relationship diagram, or data model diagram.

As to the question which grammars are used in combination, the premise of minimal ontological overlap suggests that users will use a set of grammars that together achieve maximal ontological completeness, whilst seeking to keep the set of grammars with similar representational abilities (i.e., grammars that cover the same ontological constructs) at a minimum. Grammars that cover a large extent of same ontological constructs are ontologically redundant, i.e., they provide more options to articulate a certain real-world phenomena than required, thereby unnecessarily increasing the complexity of the modeling endeavor and potentially leading to confusion as to which grammar is best to use (Weber 1997). Accordingly, our contention is that users of Excelerator will use a combination of grammars that together provide minimal ontological overlap whilst providing a maximum provision of representations. Formally:

H3 A modeler will use a combination of modeling grammars that provides maximal ontological completeness and minimal ontological overlap.

To identify all potential minimally ontologically overlapping combinations of grammars, for each starting grammar proposed above, we inspect Table 4 to determine the grammars that need to be combined with the starting grammar to make it maximally ontologically complete. For example, inspection of Table 4 shows that users of structure charts (STC) can make this grammar maximally ontologically complete whilst maintaining minimal ontological overlap, by combining STC with either the ERA or DMD grammar. Both combinations allow STC users to achieve MOC (22 ontological construct representations, as per Table 2) with an ontological overlap of 7 (as per Table 3). In a similar manner we identify MOC/MOO combinations for each of the other grammars identified in hypothesis H2 (SF, LDFD, STC, ERA, DMD).We re-write hypothesis H3 as:

H3a If an analyst/designer starts a modeling task with a system flowchart, then he/she will combine the system flowchart with a program flowchart, a structure diagram, an entity-relationship diagram or a data model diagram.

H3b If an analyst/designer starts a modeling task with a logical data flow diagram, then he/she will combine the logical data flow diagram with a program flowchart, a structure diagram, an entity-relationship diagram or a data model diagram.

H3c If an analyst/designer starts a modeling task with a structure chart, then he/she will combine the structure chart with either an entity-relationship diagram or a data model diagram.

H3d If an analyst/designer starts a modeling task with an entity-relationship diagram, then he/she will combine the entity-relationship diagram with either a program flowchart or a structure diagram.

H3e If an analyst/designer starts a modeling task with a data model diagram, then he/she will combine the data model diagram with either a program flowchart or a structure diagram.

4 Research method

We collected empirical data from users of the popular automated analysis and design tool Excelerator V1.9 using a field survey instrument. We selected the survey research method because it facilitates rigorous hypothesis testing through a sample size bigger than, for example, case studies (Gable 1994). Also, survey research has the potential to produce generalizable results that can be applied to populations other than the sample tested (King and He 2005). This approach can be of benefit to the present study to draw conclusions about the use of multiple grammars in automated tool environments other than Excelerator. Pinsonneault and Kraemer (1993) state that survey research is appropriate when clearly identified independent and dependent variables exist, and a specific model is present that theorizes the relationships between the variables. This situation is given in our hypotheses and therefore justifies the selection of the survey approach in the present study.

To minimize potential bias due to the use of one data collection instrument only, we decided to also conduct interviews to allow for the triangulation of results between the survey and the interviews, to provide deeper insights into the situation, and to overcome mono-method bias that would result from the use of a single data collection method only (Cook and Campbell 1979). We conducted the data collection as follows:

First, we designed a survey instrument to measure grammar use in Excelerator following ontological considerations. Due attention was given to the guidance provided by Straub (1989). The instrument consisted of three parts:

- a coversheet with instructions, explanations about the study and participation incentives;
- an introduction section providing important concept and term definitions; and
- a section capturing demographic information (part A) and a section capturing data about how analysts/designers used modeling grammars within Excelerator (part B).

The Appendix lists the final instrument used. Table 5 explains how the items listed in the Appendix relate to the theoretical considerations leading to the hypotheses above.

Theoretical Construct	*Survey Question*
Use of grammars in combination	B1
Recognition of manifestations of ontological incompleteness	B4a, B4c, B5a, B5c
Minimal Ontological Overlap (MOO)	B2, B3
Impact of organizational modeling standards	B4b, B4c, B5b, B5c, B6
Predictions for the use of grammars in combination	B2, B3

Table 5. Theoretical constructs and relevant survey questions

The survey instrument was pre- and pilot-tested with postgraduate students, to obtain initial feedback about validity and reliability. 437 questionnaires were sent to Excelerator users in 174 companies throughout Australia, New Zealand, and South-East Asia. One hundred and sixty-eight usable responses (a 46.5 percent response rate) resulted.

Second, we designed a structured interview protocol to follow up on the survey results with a selected subset of survey respondents. The interview protocol was crafted using the survey instrument as a basis for construction, with added prompts to each section to facilitate a conversational style; and with open-ended questions to allow interviewees to explain their reasoning for each survey question in various answers, or to enable additional answers to those provided.

The interview instrument was pilot-tested with two survey participants. After minor corrections, 34 recorded, interviewer-administered, structured interviews were performed. The interviews were later transcribed. A copy of the interview instrument and the typed transcripts of the interviews are available from the authors on request.

5 Results

5.1 Hypothesis testing

Our data analysis concerned the examination of each of our hypotheses using the data collected from the survey and the structured interviews.

With regard to the survey respondents, 66 percent had more than 5 years experience in modeling. Sixty-six percent also had greater than one year's experience with the specific modeling tool, while 25 percent indicated they had over 3 years experience with the tool at the time of the survey. Modelers of commercial systems dominated the responses over modelers of scientific/engineering systems.

We first examine Hypothesis H1. Relevant descriptive results are reported in Table 6.

Theoretical Construct	*Survey results (N = 168)*			*Interview results (N = 34)*		
	Value	*Count*	*%*	*Value*	*Count*	*%*
Use of grammars in combination	No	33	20	No	5	15
	yes	135	80	yes	29	85
	Missing	-	0	Missing	-	0
Recognition of ontological incompleteness	No	72	43	No	10	29
	yes	85	51	yes	24	71
	Missing	11	6	Missing	-	0

Table 6. Survey and interview results: Descriptive statistics

Inspection of Table 6 shows that 80 percent (135/168) of the survey respondents and 85 percent (29/34) of the interviewees used combinations of grammars. A bi-variate chi-square test of association found a significant association between the recognition of manifestation of ontological incompleteness and the use of a combination of grammars in the survey results (χ^2=7.65, p<0.01) and in the interview results (χ^2=19.63, p<0.0001). Accordingly, H1 is supported, indicating that recognition of manifestation of ontological incompleteness in the grammars provided in Excelerator is a significant influence in the decision to use multiple grammars in combination.

Examining Hypothesis H2, we identified 124 valid cases of combined-grammar use in the survey data and 29 such cases from the interviews. Table 7 presents a summary of the survey results about the grammar combinations in use, for the predicted starting grammars *viz.* SF, LDFD, STC, ERA and DMD (represented in columns). Table 7 shows, for instance, that 14 cases started their modeling with SF, of which 2 cases combined the use of SF with PF. The most popular grammar combination for SF users was LDFD (13 out of 14 cases). The rightmost column of Table 7 shows the percentage of use of the particular grammar, indicating that the LDFD grammar is the most frequently used grammar (96.7 per cent).

	First Grammar Selected										
Second Grammar Selected	*SF*		*LDFD*		*STC*		*ERA*		*DMD*		*% Use*
	Poss.	*No*	*Poss.*	*No*	*Poss.*	*No*	*Poss.*	*No*	*Poss.*	*No*	
SF	14	14	56	14	5	1	30	8	18	6	30.9
PF	14	2	56	6	5	1	30	2	18	0	8.9
LDFD	14	13	56	56	5	4	30	28	18	18	96.7
STC	14	4	56	23	5	5	30	11	18	4	38.2
TRD	14	4	56	9	5	0	30	8	18	2	18.7
STD	14	3	56	4	5	0	30	4	18	4	12.2
SDT	14	1	56	4	5	0	30	0	18	1	4.9
ERA	14	7	56	31	5	2	30	30	18	5	61
DMD	14	7	56	31	5	1	30	9	18	18	53.7
other	14	1	56	0	5	1	30	1	18	1	3.3

Table 7. Survey results: Reported grammar combinations

Table 7 shows that of the 124 valid survey cases, only one case (less than one percent) did not commence its modeling efforts with one of the five predicted starting grammars. Similarly, with the interview results, 28 out of 29 valid cases (96.5 percent) started their modeling efforts with one of the five predicted starting grammars. Accordingly, H2 appears to be supported, indicating that modelers start their modeling efforts with a grammar that provides representations for ontological system-related constructs.

Hypothesis H3 predicted combinations of grammar used on basis of MOC and MOO principles. Table 8 summarizes the number of correct predictions in relation to the reported cases of grammar combinations, for each of the hypotheses H3a-H3e.

Only six percent (7/123) of the survey cases and four percent (1/28) of the structured interview cases were predicted successfully according to any of H3a to H3e. Therefore, due to apparent lack of support for H3a to H3e, hypothesis H3 appears not to be supported. This finding indicates that maximum ontological completeness and minimum ontological overlap appear not to influence grammar combination decisions, despite influencing the starting grammar chosen. To examine this situation further, we turn to analysis of the qualitative data.

Hypothesis	*Predicted Combinations Survey*			*Predicted Combinations Structured Interviews*			*Support*
	Poss.	*No*	*%*	*Poss.*	*No*	*%*	
H3a (SF)	14	0	0	1	0	0	No
H3b (LDFD)	56	5	9	13	1	8	No
H3c (STC)	5	0	0	0	0	0	No
H3d (ERA)	30	1	3	9	0	0	No
H3e (DMD)	18	1	6	5	0	0	No

Table 8. Survey results: Prediction of grammar combinations

5.2 Post-hoc analysis

Our initial data analysis suggested no apparent support for hypothesis H3. We identify two potential explanations for this finding: (1) it could be that MOC and MOO considerations, if existent, are overwhelmed by other, confounding factors - such as individual difference factors, task factors and/or social agenda factors (Wand and Weber 2002); or (2) it could be that the MOC and MOO premises indeed do not inform grammar usage choices.

To examine these alternative explanations, we consider the qualitative responses obtained through the structured interview phase of our study, with the view to evaluate:

1. how modelers evaluate whether to add another grammar into their resultant combinations, and
2. other individual, task, and/or contextual factors that may overwhelm the influence of MOC and MOO on the resultant combinations used by analysts/designers.

Upon inspection of the reported grammar combinations (see Table 7), we noticed that a noticeable number of participants (10 out of 34) nominated the LDFD and one or other of the two data modeling grammars, ERA or DMD, as the only combination of grammars used. Our ontological analysis of these two grammars suggests that in this grammar combination, two ontological constructs cannot be represented: mono-property *state laws* and *lawful transformations*. These two ontological constructs underlie representations in conventional systems analysis for individual business rules (Recker et al. 2010), and they assist in defining representations for the lawful state space and lawful event space of classes of entities.

To uncover which means were employed by modelers using the LDFD and ERA or DMD grammar to record important business rules, we consider the qualitative responses. Table 9 summarizes the responses received, and classifies the responses into different effects.

ID	*Response*	*Effect*
SI25	We are using a separate diagramming tool to diagram business rules, business constraints. It's a separate tool to EXCELERATOR. No, there's no integration of the two. Somebody has to do that manually.	Separate tool, diagrams, no integration.
SI26	We tend not to, but we use BACHMAN now. EXCELERATOR didn't do that very well. BACHMAN does.	Use tool that can represent rules.
SI27	We didn't. No rules were recorded in EXCELERATOR.	
SI28	They were put in a separate document. Oh no... I think we recorded in the tool but it would have been as free text.	Free text.
SI29	I think they were recorded separately to EXCELERATOR. In some sort of manual fashion. It was up to someone else to remember to integrate them down the track.	Separate document, free text.
SI30	We used EXCELERATOR in conjunction with WORD (Microsoft) so we could extract the diagrams from EXCELERATOR. We would put them into a WORD document, and then we would put in the description of entities, relationships, integrity rules, volumes, etc.	Separate tool, free text.
SI31	In EXCELERATOR, it would be an English free text statement inside the process symbol definition.	Free text.
SI33	We have created a data type (in EXCELERATOR) called Business Rule, and it's just free text.	Separate type, free text
SI34	There was a Business Rule type created. We started putting in business rules individually and then cross-referencing them. The problem we have is naming them because essentially we had the same rule under many different names.	Separate type, free text.
SI35	There is an area for business rules. All there is, is the identifier and the rule in free text. This then gets uploaded to the host repository; once they get in there, they stay there and become dormant. They are sometimes documented in WORD.	Separate type, free text, separate document.

Table 9. Structured interview results: Representation of business rules

Inspection of Table 9 suggests that in 7 out of 10 cases, business rules were recorded simply using free text. This data suggests that if free-text types were used predominantly to record business rules, then a combination of LDFD, ERA or DMD, and a free-text type would be MOO-conforming. Reviewing all reported grammar combinations, such a combination accounts for up to 40 of the 124 (32 percent) combinations of grammars examined in the survey responses and up to 38 percent of combinations reported in the structured interviews. Such a situation would then lend reasonable support to the influence of MOC and MOO on which grammars are combined for use in an automated tool environment.

We further re-examined the survey and interview results to establish whether or not organizational modeling standards were in place that would mandate the use of certain grammar combinations (questions B4b, B5b, B6 in the Appendix), over and above individual grammar choices preferred by the analysts/designers. Such standards could, for example, mandate the use of more than two grammars in an analysis/design effort even if, from an ontological viewpoint, maximum ontological completeness is already achieved with two grammars. Table 10 presents two responses received on the use of standards.

ID	*Response*	*Effect*
SI24	The XXX methodology. That specifically uses two different data modelling techniques in conjunction with process modelling. They have a conceptual model which is a business model using E-R MERISE notation, then a physical data model diagram.	Standards, DB independence.
485	User requirements were for DFD, LDM, and physical DB descriptions. DMD's were preferred for one use, but ERA was required because of the limitations of the PREDICT (Corporate Repository) gateway link.	Integrate with corporate repository.

Table 10. Structured interview and survey results: Standards influencing the choice of grammar combinations

We examined the reported grammar combinations to determine if the grammar combinations predicted in H3a to H3e appeared as a subset of the reported grammar combinations, and cross-referenced the findings to the results obtained about the use of organizational standards. And indeed, in the situation of the organizational standards removed, we found that in 32 out of the 123 reported cases, the predicted grammar combinations manifested, thereby accounting for 26% of all cases. Contrasting this result to Table 8, we note a significant increase in successfully predicted grammar combinations of 18 cases (15 per cent) when controlling for the use of organizational standards. This finding, albeit not unequivocal, provides some additional evidence that MOC and MOO influence, at least partly, the decision of which grammars to use in combinations.

In summation, this qualitative evidence suggests that almost 60 percent of the survey data may not be comprehensively reflecting the influence of MOC and MOO on grammar combination choices due to the existence of contextual confounding (such as organizational standards, and personal choices to use unformalized means such as free text). We interpret this finding as suggesting that MOC and MOO are potentially valid models to explain grammar use within automated modeling tools yet confounded by task and organizational factors not controllable in the chosen research design. This situation suggests that a more controlled research design (for example, an experiment) is required to allow the influence of MOC and MOO to evidence itself.

6 Conclusions

6.1 Contributions

This research has demonstrated the application of an extended form of Wand and Weber's (1993) theory of ontological expressiveness, incorporating overlap analysis, to traditional and structured analysis modeling grammars implemented in an automated tool. The results found a strong association between recognition of manifestation of ontological incompleteness in the grammars provided by the tool and the decision to use a combination of grammars. Moreover, the theory significantly predicted the subset of five grammars from which users would select their starting grammar for modeling. While MOO/MOC alone did not appear successful in predicting which grammars an analyst/designer might use, it did predict that other means of modeling would be used in these instances. Through the additional evidence from the structured interviews we noted that where ontological incompleteness in a combination of grammars was small (e.g., one or two representational constructs were missing), end users did not seek an additional modeling grammar but instead used 'free text' types in their grammar combination to overcome the deficit.

Our results of the application of MOO/MOC in combination with data gathered from both qualitative and quantitative sources appear to support the extended theory as a fruitful avenue for research into the combined use of modeling grammars. Our findings specifically provide insights into the boundaries of the explanatory power of minimal ontological overlap, and further uncover the presence of important contextual factors (e.g., the use of organizational standards, the personal design decisions) that influence grammar usage.

6.2 Implications

For research

Our results have three main implications for research.

A first research direction flows from our extension of Wand and Weber's theory of ontological expressiveness to address multiple grammars in combination. The two premises of maximal ontological completeness and minimal ontological overlap have many potential uses for the investigation of the use of conceptual modeling in IS analysis and design. Our study provides evidence that, on the whole, the theoretical premises hold in modeling practice, and thus can be used to guide future empirical studies into modeling for IS analysis and design.

Second, this study reports on the use of qualitative research methods (i.e., structured interviews) in association with Wand and Weber's theory. Most studies to date (e.g., Recker et al. 2011; Shanks et al. 2008) have used quantitative inquiries (most notably, surveys and experiments). Our work shows that qualitative inquiry can be used to extend the scope and nature of inquiry into conceptual modeling practice and inform IS scholars on why – or why not – theoretical premises hold in real life modeling situations. We believe that future studies following a dedicated qualitative paradigm can meaningfully extend our work and provide further insights into conceptual modeling practice.

Third, we have provided an extended operationalization of the concepts of maximum ontological completeness and minimal ontological overlap, and provided an initial instrument that allows researchers to measure the manifestation of MOC and MOO in conceptual modelling practice. Future research may now further advance our instrument in other studies on how end users combine multiple grammars.

For practice

We identify at least three significant implications for practice. First, our findings can be used to guide modeling grammar (re-) development. Specifically, developers of conceptual grammars should pay attention to the levels of ontological incompleteness a grammar exhibits. Our study showed that, upon recognition of ontological incompleteness, modelers seek additional means to aid their modeling – be it in the form of other grammars, or through the use of non-formalized textual annotations. The design of grammars that are ontologically more complete could be of substantial assistance to end users in that they will not have to rely on non-formalized textual means to specify information systems requirements.

Second, the implementation of multiple grammars in modeling tools should be performed with a view of eliminating ontological deficiencies across the implemented set of grammars. Specifically, tool developers should consider the maximum level of ontological completeness made available through the grammars. They should further choose grammars for implementation that share only minimal ontological overlap, so as to reduce additional complexity costs that would arise from using multiple grammars that share similar representational capacities.

Third, users of a modeling tool should be aware of the ontological incompleteness of the grammars implemented within the tool and should have an appreciation of which grammars can be best combined to achieve their modeling purpose with minimal construct overlap. This could be achieved, for instance, through appropriate training or modeling conventions.

6.3 Limitations

We note that our analysis and the ensuing empirical study is based on the representation mapping of grammatical constructs to the ontological constructs specified by Wand and Weber (1990) in their ontological model. As has been discussed extensively (Kautz et al. 2006), Wand and Weber's model is one potential ontology against which modeling grammars might be evaluated, and fellow scholars may or may not subscribe to the viewpoints expressed in their model. Our work complements and extends existing research on the basis of Wand and Weber's (1990) model (e.g., Bodart et al. 2001; Bowen et al. 2009), and it provides some evidence that the use, adaptation, and extension of Bunge's ontological considerations can be successfully employed to acquire insights into domains, procedures and outcomes of conceptual modeling in information systems practice.

Second, both representation and interpretation mapping are inherently reliant on the subjective assessment by the researcher concerned with the exercise. The subjective interpretation bias, therefore, may threaten the validity of the analysis results. One potential mitigation mechanism could be the involvement of an expert panel to review the mapping results. Such an approach, however, was precluded in our study for the pragmatic reasons of availability, time and cost. In Section 3.2 we report, however, on the conduct of mapping, and the rationale behind mapping for the ERA grammar, to illustrate our line of reasoning to the reader. We also refer to the example of a multi-coder mapping approach described in (Recker et al. 2010) as another example for increasing the mapping reliability.

Last, we studied responses from end users, gathered through a survey and structured interview. We note that an examination of how modelers use modeling grammars could also make use of participatory observation (Nandhakumar and Jones 2002), verbal protocol analysis (Purao et al. 2002) or similar data analysis techniques that allow for deeper insights into the process of using conceptual modeling grammars in combination. We invite fellow scholars to extend our research by considering such approaches.

7 References

Bodart, F., Patel, A., Sim, M., and Weber, R., (2001). Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. *Information Systems Research*, (12:4): 384-405.

Bowen, P. L., O'Farrell, R. A., and Rohde, F., (2009). An Empirical Investigation of End-User Query Development: The Effects of Improved Model Expressiveness vs. Complexity. *Information Systems Research*, (20:4): 565-584.

Bunge, M. A., (1977). *Treatise on Basic Philosophy Volume 3: Ontology I - The Furniture of the World*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Burton-Jones, A., and Meso, P., (2006). Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis. *Information Systems Research*, (17:1): 38-60.

Chau, P. Y. K., (1996). An Empirical Investigation on Factors Affecting the Acceptance of CASE by Systems Developers. *Information & Management*, (30:6): 269-280.

Cook, T. D., and Campbell, D. T., (1979). *Quasi-Experimentation: Design and Analysis Issues*, Houghton Mifflin, Boston, Massachusetts.

Fettke, P., (2009). How Conceptual Modeling Is Used. *Communications of the Association for Information Systems*, (25:43): 571-592.

Gable, G. G., (1994). Integrating Case Study and Survey Research Methods: An Example in Information Systems. *European Journal of Information Systems*, (3:2): 112-126.

Green, P., (1997). Use of Information Systems Analysis and Design (ISAD) Grammars in Combination in Upper CASE Tools - An Ontological Evaluation. In: *2nd CAiSE/ IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, University of Nebraska-Lincoln, Barcelona, Spain, pp. 1-12.

Green, P., and Rosemann, M., (2004). Applying Ontologies to Business and Systems Modeling Techniques and Perspectives: Lessons Learned. *Journal of Database Management*, (15:2): 105-117.

Green, P., Rosemann, M., Indulska, M., and Manning, C., (2007). Candidate Interoperability Standards: An Ontological Overlap Analysis. *Data & Knowledge Engineering*, (62:2): 274-291.

Iivari, J., (1996). Why are CASE Tools not Used? *Communications of the ACM*, (39:10): 94-103.

Kautz, K., Wyssusek, B., Lyytinen, K., Milton, S., Kazmierczak, E., Opdahl, A. L., Krogstie, J., Guarino, N., Guizzardi, G., Wand, Y., and Weber, R., (2006). Debate Section: On Ontological Foundations of Conceptual Modelling. *Scandinavian Journal of Information Systems*, (18:1): 61-152.

King, W. R., and He, J., (2005). External Validity in IS Survey Research. *Communications of the Association for Information Systems*, (16:45): 880-894.

Martin, G., and De Luca, F., (). CASE Usage in Australia - Survey Summary. Monash University, Melbourne, Australia.

Moody, D. L., (2005). Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. *Data & Knowledge Engineering*, (15:3): 243-276.

Nandhakumar, J., and Jones, M., (2002). Development Gain? Participant Observation in Interpretive Management Information Systems Research. *Qualitative Research*, (2:3): 323-341.

Pinsonneault, A., and Kraemer, K. L., (1993). Survey Research Methodology in Management Information Systems: An Assessment. *Journal of Management Information Systems*, (10:2): 75-105.

Premkumar, G., and Potter, M., (1995). Adoption of Computer Aided Software Engineering (CASE) Technology: An Innovation Adoption Perspective. *Database for Advances in Information Systems*, (26:2-3): 105-123.

Purao, S., Rossi, M., and Bush, A., (2002). Towards an Understanding of Problem and Design Spaces during Object-oriented Systems Development. *Information and Organization*, (12:4): 249-281.

Recker, J., (2011). *Evaluations of Process Modeling Grammars: Ontological, Qualitative and Quantitative Analyses Using the Example of BPMN*, Springer, Berlin.

Recker, J., Indulska, M., Rosemann, M., and Green, P., (2010). The Ontological Deficiencies of Process Modeling in Practice. *European Journal of Information Systems*, (19:5): 501-525.

Recker, J., Rosemann, M., Green, P., and Indulska, M., (2011). Do Ontological Deficiencies in Modeling Grammars Matter? *MIS Quarterly*, (35:1):57-79.

Ricciuti, M., (1992). Can Intersolv Excelerate on OS/2? *Datamation*, (March): 24-27.

Shanks, G., Tansley, E., Nuredini, J., Tobin, D., and Weber, R., (2008). Representing Part-Whole Relations in Conceptual Modeling: An Empirical Evaluation. *MIS Quarterly*, (32:3): 553-573.

Sørensen, C., (1993). What Influences Regular CASE Use in Organizations?: An Empirical Based Model. *Scandinavian Journal of Information Systems*, (5:1): 25-50.

Straub, D. W., (1989). Validating Instruments in MIS Research," *MIS Quarterly*, (13:2): 147-169.

Vessey, I., Jarvenpaa, S. L., and Tractinsky, N., (1992). Evaluation of Vendor Products: CASE Tools and Methodology Companions. *Communications of the ACM*, (35:4): 90-105.

von Halle, B., (2001). *Business Rules Applied: Building Better Systems Using the Business Rules Approach*, John Wiley & Sons, New York.

Wand, Y., and Weber, R., (1990). An Ontological Model of an Information System. *IEEE Transactions on Software Engineering*, (16:11): 1282-1292.

Wand, Y., and Weber, R., (1993). On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems*, (3:4): 217-237.

Wand, Y., and Weber, R., (1995). On the Deep Structure of Information Systems. *Information Systems Journal*, (5:3): 203-223.

Wand, Y., and Weber, R., (2002). Research Commentary: Information Systems and Conceptual Modeling - A Research Agenda. *Information Systems Research*, (13:4): 363-376.

Weber, R., (1997). *Ontological Foundations of Information Systems*, Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia.

Yourdon, E., (1989). *Modern Structured Analysis*, Prentice-Hall, Upper Saddle River, New Jersey.

8 Appendix

Survey instrument (excerpt).

A. Background

1. I have been analyzing/designing computer systems:
   a. < 2 years
   b. 2-5 years
   c. 5-15 years
   d. 15 years
2. I have used, or have been using, Excelerator for:
   a. < 1 year
   b. 1-3 years
   c. 3 years
3. The category of computer system that I most normally analyse/design is:
   a. Commercial
   b. Engineering/scientific
4. My organization would be classified as:
   a. Public sector
   b. Private sector

B. Use of Grammars in Excelerator

1. When analyzing/designing a computer system in Excelerator, I use more than one modeling grammar:

   a. Yes
   b. No

2. As I use more than one modeling grammar when analyzing/designing a system, I normally begin my analysis/design effort with the modeling grammar (select one)
   a. System Flowchart (PRG)
   b. Program Flowchart (PRG)
   c. Data Flow Diagram (DFD)
   d. Structure Chart (STC)
   e. State Transition Diagram (TRD)
   f. Structure Diagram (STD)
   g. Structure Decision Table (SDT)
   h. Entity-Relationship Diagram (ERA)
   i. Data Model Diagram (DMD)
   j. Other, please specify
3. Having started my analysis/design with the technique nominated above, I then normally use the following techniques to complete the analysis/design (select all that apply):
   a. System Flowchart (PRG)
   b. Program Flowchart (PRG)
   c. Data Flow Diagram (DFD)
   d. Structure Chart (STC)
   e. State Transition Diagram (TRD)
   f. Structure Diagram (STD)
   g. Structure Decision Table (SDT)
   h. Entity-Relationship Diagram (ERA)
   i. Data Model Diagram (DMD)
   j. Other, please specify
4. I use more than one modeling grammar in Excelerator to analyze/design a computer system because (yes/no for each sub-question):
   a. There is no one modeling grammar within Excelerator that has a symbol for all the concepts I need to represent in a complete analysis/design.

   b. That combination of modeling grammars is the standard set to be used within my organization.

   c. Other, please specify.

5. I use only one modeling grammar in Excelerator to analyze/design a computer system because (yes/no for each sub-question):

   a. There are sufficient symbols in my selected grammar for all the concepts I need to represent in a complete analysis/design.

   b. That is the standard procedure set by my organization.

   c. Other, please specify.

6. For analysis and design work using Excelerator, my organization requires that I use a standard set of grammars/symbols (yes/no).