An Ontological Analysis of Integrated Process Modelling

Peter Green¹, Michael Rosemann²

¹ University of Queensland, Department of Commerce
Salisbury Road, Ipswich QLD 4035, Australia
p.green@mailbox.uq.edu.au

² University of Muenster, Department of Information Systems
Steinfurter Str. 107, D-48149 Muenster, Germany
ismiro@wi.uni-muenster.de

Abstract. Process modelling has gained prominence in the information systems modelling area due to its focus on business processes and its usefulness in different business improvement methodologies. However, process modelling techniques are not without their criticisms. This paper proposes and uses the Bunge-Wand-Weber (BWW) representation model to analyse the four views - process, data, function, and organisation - provided in the Architecture of Integrated Information Systems (ARIS) popularised by Scheer [39, 40, 41]. The BWW representation model attempts to provide a theoretical base on which to evaluate and thus contribute to the improvement of information systems modelling techniques. The analysis conducted in this paper prompts some propositions. Among others, it confirms that the process view alone is not sufficient to model all the real-world constructs required. However, even when considering all four views in combination, problems may arise in representing all potentially required business rules, and specifying the scope and boundaries of the system under consideration, and employing a “top-down” approach to analysis and design.

1 Introduction

Methodological issues surrounding information systems development - the analysis, design, construction and implementation tasks - have long been central to the interest of information systems professionals, practitioners and researchers alike. Hirschheim *et al.* [19, 20], and most recently, Iivari *et al.* [22], and Mylopoulos [29] have reviewed rigourously many methodologies and their underlying philosophies as applied to information systems development. Researchers in information systems development have for years lamented the fact that little theoretical guidance only has been provided to practitioners on several areas involved in IS development [2, 15]. Consequently, methodologies, techniques, and grammars¹ have proliferated over time

¹ The terms methodology, technique, and grammar are distinguished in the following way in this paper. A *methodology* provides a comprehensive approach to systems planning, analysis, design, and construction such as Andersen’s Method/1. It may include one or more techniques. A *technique* such as data flow diagramming designates a set of concepts and a way of handling them. Within a

[24, 30]. This situation has contributed to what Banville and Landry [3] describe as the “fragmented adhocracy” of the state of theoretical development in the IS discipline. By contrast, Benbasat and Weber ([6], p. 398) implore the IS community to “not mix up the notions of the *core* of the IS discipline and the *body of knowledge* for the IS discipline.” Moreover, Benbasat and Weber [6] go on to advocate that diversity (adhocracy) clearly has its place in IS research but not as an excuse for shirking the fundamental responsibility of a community to build its own theories to account for those *core* phenomena that differentiate the IS discipline from other disciplines.

In an attempt to address this situation, Wand and Weber [47, 48, 49, 50, 51, 52, 53] have developed and refined a set of models that specify what they believe are a set of core phenomena for the IS discipline. These models are based on an ontology defined by Bunge [7] and are referred to as the Bunge-Wand-Weber (BWW) models. These models, in particular their representation model, provide a theoretical basis on which information systems modelling grammars and the scripts prepared using such grammars can be evaluated. As Weber [54] argues, this evaluative aspect of the models persists irrespective of the philosophical assumptions under which the models are applied. His and Wand’s central concern is with the goodness of the *representation* of the perception of that portion of the real world that is being modelled. Various researchers have demonstrated the applicability and usefulness of these models for such grammars as data flow diagrams, entity-relationship diagrams, object-oriented schemas, the relational model, NIAM, and grammars in CASE tools [17, 31, 44, 46, 48, 52, 53, 54, 55].

For many years now however, there has been an increased recognition in information systems modelling of the dynamic behaviour of organisations. Process modelling has been embraced as an appropriate approach to describe the behaviour and as a mechanism by which many of the related concerns with the traditional modelling grammars can be overcome [10]. Moreover, process modelling focuses on understanding the underlying business processes which many IS professionals believe is fundamental to the successful implementation of technology-based change in organisations [13]. As Becker *et al.* ([5], p. 821) explain, “process models are ... images of the logical and temporal order of functions performed on a process object. They are the foundation for the operationalisation of process-oriented approaches.”

The popularity of concentrating on business processes through process modelling has been fuelled over the last ten years by the prominence of such organisational improvement approaches as Total Quality Management, Time-based Management, Business Process Reengineering (BPR), Value-based Performance Measurement, and Workflow Management. Furthermore, the rapid worldwide take-up over the last five years of enterprise resource planning (ERP) software such as SAP R/3 or BaanERP that rely heavily on reference process models to describe the software functionality and to guide and document implementation has added significantly to the interest and

technique, these concepts are represented typically by formal graphical symbols. In this paper, the set of symbols together with its construction rules is called a *grammar*.

perceived usefulness of process modelling. In particular in conjunction with the success of software specific reference models, integrated process modelling that provides various *views* of the process to users has received much attention [9, 14, 28]. However, process modelling is not without its critics who point to such deficiencies as an emphasis on the “hard” factors (who, what, when, and where) to the detriment of the “soft” factors (norms, beliefs, and motives) in the examination and modelling of processes [13].

The aim then of this study is to analyse as an example the integrated process modelling grammar within the ARIS-framework popularised by Scheer [39, 40, 41] using the BWW theory base to determine the ability of this grammar to provide “good representations of the perceptions” of business analysts. Accordingly, this work is motivated by several factors. First, through such an analysis, potential weaknesses of integrated process modelling grammars can be identified. Such an analysis can then potentially contribute to the theoretical development of integrated process modelling. Second, the results of the analysis in this paper may be useful to the implementers and users of comprehensive process-based software systems such as SAP R/3 or BaanERP. SAP has stated that in subsequent versions of its product they intend the implementation/ customisation process to be driven directly by modifications made to the relevant reference process models supplied in the product. In BaanERP it is already possible to derive from the tailored reference model the information necessary for system customisation and authorisation ([45], p. 4). In such a situation, potential weaknesses in integrated process modelling identified in this analysis may provide useful information to the implementers of such software systems. Third, this study provides an opportunity to extend the existing work on the application of the BWW models. To date, the BWW models have in the main been applied to “traditional” information systems analysis and design (ISAD) grammars. This paper extends their application into the dynamic area of process modelling. Fourth, this study will add to the development of the BWW models by extending their application to a different modelling environment. By such a further application, the robustness of the BWW ontological constructs can be examined. Finally, the implications of the analysis can be articulated as hypotheses and empirically tested with a base of integrated process modelling users. The results of such a step will contribute to the development of the BWW models and integrated process modelling.

Consequently, this paper proceeds in the following manner. First, some background on the development of the BWW models and integrated process modelling is given. This section is followed by the analysis of the integrated modelling grammars. Included in this section is a brief discussion of why Scheer’s Architecture of Integrated Information Systems was selected for analysis. The analytical methodology is explained using the process view of this integrated modelling grammars as an example. In particular, the ontological implications of examining the views in combination are discussed. Following this work, an initial set of hypotheses is presented. Finally, some conclusions and directions for further research are sketched.

2 Background

2.1 The Bunge-Wand-Weber (BWW) Models

As the number of grammars for information systems modelling has grown over the years, information systems researchers and professionals together have attempted to derive bases on which to compare, evaluate, and determine when to use these grammars [4, 24, 43]. Since the advent of computers, grammars for information systems requirements modelling have progressed from flowcharts, to data flow diagrams, to entity-relationship diagrams, to object-oriented schemas and integrated process modelling systems [29]. Since the early 1980's, however, many IS researchers have lamented the paucity of theoretical foundation for modelling grammar specification.

In an attempt to provide such a theoretical foundation for the design and the evaluation of modelling grammars, Wand and Weber [47, 48, 49, 50, 51, 52, 53] have investigated the branch of philosophy known as ontology (or meta-physics) as a theoretical foundation for understanding the modelling of information systems. Ontology is a well-established theoretical domain within philosophy dealing with models of reality, that is, the nature of the real world ([29], 30-33). Wand and Weber [48, 49, 50, 52, 53] have taken, and extended, an ontology presented by Bunge [7] and applied it to the modelling of information systems. Their fundamental premise is that any modelling 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.

The Bunge-Wand-Weber (BWW) [48, 49, 50, 52, 53] models consist of the representation model, the state-tracking model, and the good decomposition model. This work concentrates on the use of the representation model. The representation model defines a set of constructs that, at this time, are thought by Wand and Weber to be necessary and sufficient to describe the structure and behaviour of the real world. To date, however, concerns remain, *inter alia*, with regard to a lack of understandability of the BWW constructs, the problematic application of these constructs to other loosely defined modelling grammars, and the limited empirical testing of the implications of the BWW models [44, 52, 54]. This work attempts to mitigate these concerns in part. Table 1 presents definitions for the constructs of the BWW representation model.

Table 1. Ontological constructs in the BWV Representation model

Ontological Construct	Explanation
THING*	A thing is the elementary unit in the BWV ontological model. The real world is made up of things. Two or more things (composite or simple) can be associated into a composite thing.
PROPERTY*: -IN GENERAL -IN PARTICULAR -HEREDITARY -EMERGENT -INTRINSIC -NON-BINDING MUTUAL -BINDING MUTUAL -ATTRIBUTES	Things possess properties. A property is modelled via a function that maps the thing into some value. For example, the attribute "weight" represents a property that all humans possess. In this regard, weight is an attribute standing for a property in general . If we focus on the weight of a specific individual, however, we would be concerned with a property in particular . A property of a composite thing that belongs to a component thing is called an hereditary property. Otherwise it is called an emergent property. Some properties are inherent properties of individual things. Such properties are called intrinsic . Other properties are properties of pairs or many things. Such properties are called mutual . Non-binding mutual properties are those properties shared by two or more things that do not "make a difference" to the things involved; for example, order relations or equivalence relations. By contrast, binding mutual properties are those properties shared by two or more things that do "make a difference" to the things involved. Attributes are the names that are used to represent properties of things.
CLASS	A class is a set of things that can be defined via their possessing a single property.
KIND	A kind is a set of things that can be defined only via their possessing two or more common properties.
STATE*	The vector of values for all property functions of a thing is the state of the thing.
CONCEIVABLE STATE SPACE	The set of all states that the thing might ever assume is the conceivable state space of the thing.
STATE LAW: -STABILITY CONDITION -CORRECTIVE ACTION	A state law restricts the values of the properties of a thing to a subset that is deemed lawful because of natural laws or human laws. The stability condition specifies the states allowed by the state law. The corrective action specifies how the value of the property function must change to provide a state acceptable under the state law.
LAWFUL STATE SPACE	The lawful state space is the set of states of a thing that comply with the state laws of the thing. The lawful state space is usually a proper subset of the conceivable state space.
CONCEIVABLE EVENT SPACE	The event space of a thing is the set of all possible events that can occur in the thing.
TRANSFORMATION*	A transformation is a mapping from one state to another state.
LAWFUL TRANSFORMATION -STABILITY CONDITION -CORRECTIVE ACTION	A lawful transformation defines which events in a thing are lawful. The stability condition specifies the states that are allowable under the transformation law. The corrective action specifies how the values of the property function(s) must change to provide a state acceptable under the transformation law.
LAWFUL EVENT SPACE	The lawful event space is the set of all events in a thing that are lawful.
HISTORY	The chronologically-ordered states that a thing traverses in time are the history of the thing.
ACTS ON	A thing acts on another thing if its existence affects the history of the other thing.
COUPLING: BINDING MUTUAL PROPERTY	Two things are said to be coupled (or interact) if one thing acts on the other. Furthermore, those two things are said to share a binding mutual property (or relation); that is, they participate in a relation that "makes a difference" to the things.
SYSTEM	A set of things is a system if, for any bi-partitioning of the set, couplings exist among things in the two subsets.
SYSTEM COMPOSITION	The things in the system are its composition.
SYSTEM ENVIRONMENT	Things that are not in the system but interact with things in the system are called the environment of the system.
SYSTEM STRUCTURE	The set of couplings that exist among things within the system, and among things in the environment of the system and things in the system is called the structure.
SUBSYSTEM	A subsystem is a system whose composition and structure are subsets of the composition and structure of another system.

Table 1. Ontological constructs in the BWW Representation model

Ontological Construct	Explanation
SYSTEM DECOMPOSITION	A decomposition of a system is a set of subsystems such that every component in the system is either one of the subsystems in the decomposition or is included in the composition of one of the subsystems in the decomposition.
LEVEL STRUCTURE	A level structure defines a partial order over the subsystems in a decomposition to show which subsystems are components of other subsystems or the system itself.
EXTERNAL EVENT	An external event is an event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment on the thing, subsystem, or system.
STABLE STATE*	A stable state is a state in which a thing, subsystem, or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event).
UNSTABLE STATE	An unstable state is a state that will be changed into another state by virtue of the action of transformations in the system.
INTERNAL EVENT	An internal event is an event that arises in a thing, subsystem, or system by virtue of lawful transformations in the thing, subsystem, or system.
WELL-DEFINED EVENT	A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known.
POORLY-DEFINED EVENT	A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known.

Source: [49, 53, 54] with minor modifications. * indicates a fundamental and core ontological construct.

2.2 Integrated Process Modelling

A process can be defined as the sequence of functions that are necessary to transform a business-relevant object (*e.g.*, purchase order, invoice). From an information systems perspective, a model of a process is a description of the control flow. Process modelling is not a new phenomenon. Indeed, the research into process modelling predates that of data modelling if Chen’s milestone article [8] about the E-R model represents the birth of data modelling. The basic grammar of most process modelling approaches derives from Petri’s doctoral thesis [33]. Since that time, intensive research on this topic has been undertaken. Today, many different, often highly sophisticated, Petri-net approaches exist [32, 34]. All in all, regarding the methodology the Petri-net related research appears to be well developed, however, it lacked a realistic business application.

At the beginning of this decade, the idea of process management gained significant prominence [11, 12, 18, 19, 42]. From a strategic business viewpoint, process management advocated a radical change from a functional focus to a (business) process orientation to cope successfully with new challenges like customer focus and technological development. The publications of Hammer [18, 19] and Davenport [11] in particular triggered intensive research on process management. Accordingly, it would seem now that a process orientation is achieving popularity not only in the business administration discipline but also within the information systems community. For example, workflow management systems [16, 23] represent a new process-oriented middleware.

Process modelling for the various purposes of process management faces three main (new) requirements. First, the process model must include more information

than just a representation of the control flow. In addition to the control flow, a comprehensive process model should include at least:

1. Information about the *organisational units* (e.g. internal and external persons, departments, (project) groups) that are involved in the process by having to execute certain tasks or be responsible for the performance of those tasks;
2. the *data* that are necessary for a function to process - input data - and the data that result from the execution of a function - output data; and
3. references to existing repositories that describe the *functions* within an enterprise. As Curtis *et al.* (1992, p. 75) explain, "Process modelling is distinguished from other types of modelling in computer science because many of the phenomena modelled must be enacted by a human rather than a machine." Thus a process model should also include manual functions.

Further requirements of business process modelling include the integration of business objectives, business partners such as customers and vendors, temporal aspects (e.g. deadlines), involved knowledge, and/or resources such as application systems.

Second, a process model should not only be comprehensive but it must also be *easy to understand*. Only in this case it can be used as a platform for communication with business people who are not familiar with process modelling. Current elaborated Petri-nets cannot fulfil this requirement as they are developed from method experts for method experts. Third, process models should be *based on a meta model* to serve as a well-defined starting point for the development of software like workflow-based applications [26]. Further applications of process models as a result of the requirements engineering that require a runtime component are animation or simulation. All in all, process models, that are not only used for the purpose of developing software must be comprehensive, understandable and formal at the same time.

3 The Analysis

3.1 Why Analyse Scheer's [39, 40, 41] Work?

The increasing prominence of process management together with insufficient existing Petri-net approaches has motivated a significant amount of research in the business process modelling area. One of the most successful grammars for process modelling is *event-driven process chains (EPC)*. This grammar is embedded in the *Architecture of Integrated Information Systems (ARIS)* [1, 25, 39, 40, 41]. Like Petri-nets, event-driven process chains are bipartite graphs with active nodes (functions, designed as soft rectangles) and passive nodes (events, designed as hexagons). In addition to the facilities of Petri-nets, event-driven process chains express explicitly the logic of the control flow, *viz.*, joining and splitting points. Three connectors are used to describe the logical AND, the exclusive OR (A or B), and the inclusive OR (A or B or (A and B)). These connectors represent a principal point of differentiation from most Petri-net based grammars. In summary then, an event-driven process chain depicts who (organisational view) is responsible for which function (functional view), what input data is needed, and what output data is produced by the transformation of which

business objects (data view). Figure 1 shows this integrated process modelling approach embedded in the so-called ARIS house and without the output view [39, 40, 41], which includes the output of every function. This view is substituted by representing the process object (here: purchase order), which is related to the data view ([35], 76-84).

Fig. 1. Integration in the Architecture of Integrated Information Systems

The integration of four views and the relatively easy-to-understand resultant process models are not the only reasons the ARIS-approach was selected to exemplify an ontological analysis of integrated process modelling. Other reasons include:

1. ARIS-Toolset 4.0 provides a “cutting-edge” upper CASE tool that supports the Architecture of Integrated Information Systems. With more than 10,000 licences worldwide, there appears to be a large, mature user base against which the implications of the analysis in this work can be tested. Furthermore, on extending the analysis in future work to cover the implementation aspects of process modelling, again a mature user base will be available for verification of hypotheses.
2. One of the world’s most popular ERP software packages, SAP R/3, also uses the grammar that is employed principally within ARIS to describe processes - the event-driven process chain [9, 28]. Within SAP R/3, more than 800 event-driven

process chains explain the functionality of the software - the so-called business blueprint. Consequently, it appears to be a relatively straightforward task to extend the ontological analysis to the software-specific reference models of this ERP-software.

First attempts at integrating event-driven process chains with object-oriented approaches such as UML have been reported [27]. Consequently, this *de facto* object-oriented language represents an area of potential interest for ongoing research stemming from the work reported in this paper.

3.2 Analysis Performed

The aim of this work is to extend the BWV ontological analysis to “state of the art” business process modelling grammars.

Table 2 presents the ontological analysis of the ARIS views (process, data, function, organisation). To explain the application of the analytical methodology, the process view of the Architecture of Integrated Information Systems will be used as an example. A *thing* as an elementary ontological construct is not a part of the original meta model of event-driven process chains.² Because a function type within an event-driven process chain can be seen as the transformation of a business relevant object, an EPC function type can be interpreted to represent a *property in general* of that object. Attribute types in EPC represent *attributes* in the BWV representation model. The ontological construct *class*, however, is not represented in the EPC-grammar.

As opposed to grammars that depict the structure of a system (*e.g.*, the Entity-Relationship (E-R) model), process modelling languages focus on the behavioural aspects of what is being modelled. Consequently, the ontological constructs *state*, *transformation*, and *event* are most relevant. *Transformations* are represented by function types in the event-driven process chains while *states* are depicted as event types. Accordingly, the triple, event type – function type – event type, in an EPC represents the ontological construct *event*, and usually *internal events* that are *well-defined*. The homonym between the EPC event type and the ontological *event* requires careful attention during the analysis. Similarly, a *state law* can be represented by the triple, function type – connector – event type, while a *lawful transformation* can be represented by the pattern, event type – connector – function type. An *external event* may be represented by the start event type at the beginning of an EPC while the final *stable state* (of an object) may be represented by the end event type at the bottom of an EPC.

² The original work of Scheer [41] is referenced here. Further approaches that extend this meta model with business objects (*e.g.*, [27, 35]) are not analysed in this study.

Table 2. BWW Representation Model Analysis of Integrated Process Modelling Views				
Ontological Construct	Process View	Data View	Function View	Organisational View
THING				User, department (instances)
PROPERTY: IN PARTICULAR IN GENERAL INTRINSIC MUTUAL EMERGENT HEREDITARY ATTRIBUTES	Function Type		Function Type	
	Attribute type	Attribute type	Attribute type	Attribute type
CLASS		Entity type		Organisation type
KIND		Specialisation/generalisation (IS-A)	Specialisation/generalisation (IS-A)	Organisation type
STATE	Event type (only the state variables that trigger the function)			
CONCEIVABLE STATE SPACE				
STATE LAW	Function type → connector → Event type	Specialisation/generalisation descriptors; [Min., max.] cardinalities	Specialisation/generalisation descriptors	
LAWFUL STATE SPACE				
EVENT	Event type → Function type → Event type			
PROCESS	Process model Function type		Function type Process oriented function decomposition	
CONCEIVABLE EVENT SPACE				
TRANSFORM-ATION	Function type		Function type	
LAWFUL TRANSFORM-ATION	Event type → connector → Function type			
LAWFUL EVENT SPACE				
HISTORY				
ACTS ON				
COUPLING: BINDING MUTUAL PROPERTY		Relationship type (no symbol for relationship in grammar)		
SYSTEM				
SYSTEM COMPOSITION		Cluster		
SYSTEM ENVIRONMENT				
SYSTEM STRUCTURE				
SUBSYSTEM		Cluster		Organisational hierarchy
SYSTEM DECOMPOSITION	Function type decomposition indicator Event type decomposition indicator			Organisational hierarchy
LEVEL STRUCTURE	Series of function type or event type decomposition indicators		Series of function type decomposition indicators	

Table 2. BWW Representation Model Analysis of Integrated Process Modelling Views

Ontological Construct	Process View	Data View	Function View	Organisational View
EXTERNAL EVENT	Start event type (no ingoing arrow)			
STABLE STATE	End event type (no outgoing arrow; event types generally have one ingoing & one outgoing arrow)			
UNSTABLE STATE				
INTERNAL EVENT	Event type → Function type → Event type			
WELL-DEFINED EVENT	Event type → Function type → Event type			
POORLY-DEFINED EVENT				

4 Some Propositions

In the light of an *isolated analysis of the process view*, some propositions can be made:

1. Because there are no direct representations for thing, class, and/or kind, users will lack conceptual clarity regarding the object(s) in the real world to which the EPC relates. The EPC can represent indirectly attributes of the thing (business object) as attributes of the function type but not the thing itself. As Weber [54] explains, the world is made up of things. Things in the world are identified via their properties; not the other way round. Accordingly, some other symbol/grammar/view will be needed in conjunction with the process view to overcome this ontological deficiency.
2. Because the process view does not have representations for *conceivable state space*, *lawful state space*, *conceivable event space*, and *lawful event space*, a sufficient focus to identify all important state and transformation laws may not be present during modelling. These laws are the basis of what are known in systems analysis as business rules. Accordingly, problems may be encountered in capturing all the potentially important business rules of the situation. Again, some other symbol/grammar/view may need to be employed in combination with the process view to overcome this shortage.
3. Because the process view does not have representations for *system*, *system composition*, *system environment*, *system structure*, *subsystem*, *history*, *acts on*, *unstable state*, and *poorly-defined event*, the process view's usefulness for defining the scope and boundaries of the system being analysed is undermined. Moreover, the usefulness of the process view for undertaking "good decompositions" (see [53], for example) during analysis is undermined. Again, the use of some other symbol/grammar/view would appear necessary to overcome this deficiency.

Scheer's [39, 40, 41] integrated process modelling is designed around the four views to reduce complexity during the analysis and design task. Accordingly, the views are

intended to be used in combination during a project. Moreover, this combined use of the views is encouraged by Scheer's [41] claims that, when determining which views to use, relationships within views are high while the relationships between the views are relatively simple and loosely linked.

When the *analysis of the four views in combination* in Table 2 is considered, some interesting propositions issue.

4. Even across the four views, no representations exist for *conceivable state space*, *lawful state space*, *conceivable event space*, or *lawful event space*. Again, a sufficient focus to identify all important state and transformation laws may not be present during modelling. Accordingly, problems may be encountered in capturing all the potentially important business rules of the situation. Now, some other symbol/grammar/view *in addition to the integrated process modelling system* may need to be employed to overcome this deficiency.
5. Across the four views, no representations exist for *system*, *system environment*, *system structure*, *history*, *acts on*, *unstable state*, and *poorly-defined event*. Again, its usefulness for defining the scope and boundaries of the system being analysed is undermined. Moreover, the usefulness of integrated process modelling for undertaking "good decompositions" during analysis would seem undermined. Indeed, because of this situation, integrated process modelling would appear more useful in modelling existing situations to be modified (*i.e.*, where a bottom-up approach is appropriate) as opposed to novel situations (*i.e.*, where a top-down approach might be more appropriate).

Of course, other explanations may exist for the fact that representations for the ontological constructs mentioned in propositions 4 and 5 do not exist in any of the four views. First, the BWW theory may be misspecified or simply wrong. This question however can only be answered after substantially more empirical testing has been performed on the models. Second, for practitioners in the real world, the cost of having and using symbols for these particular ontological constructs in the process modelling views may outweigh the representational benefits that they provide. This question refers to the economical efficiency of modelling activities [36] and is also one that must be answered with empirical evidence.

Green [17] investigated the use of information systems analysis and design (ISAD) grammars in combination within upper CASE tools. In that work, he operationalised a concept known as *minimal ontological overlap (MOO)*. This concept was proposed by Weber and Zhang [55] when they concluded from their field work that users would find most useful combinations of grammars in which the grammars overlapped minimally in terms of ontological constructs. Green [17] proposed that, to form a MOO-set of grammars, *inter alia*, one would select a starting grammar and add grammars to it until the set was as ontologically complete as possible (*maximal ontological completeness (MOC)*). Following the rule of parsimony, this set of grammars would be formed using as few grammars as possible. Applying these concepts to the views in Table 2, if the process view is selected as a starting point, then the data view and the organisational view would need to be employed to form the MOO set. The function view however appears to be ontologically redundant when compared to the MOO-set of grammars. This situation suggests the following proposition:

6. Because the function view is ontologically redundant when compared to the combination of process view, data view, and organisational view, its use with the other views will undermine clarity and may cause confusion to users. Accordingly, limited usefulness and use of the function view is expected.

5 Conclusion and Future Directions

This paper has analysed the event-driven process chain popularised by Scheer using the BWV theory base to determine the ability of this grammar to provide a “good representation of the perceptions” of business analysts. Through such an analysis, it has identified some areas where potential contributions can be made to the theoretical development of integrated process modelling. Moreover, this study has extended the existing work on the application of the BWV models. To date, the BWV models have in the main been applied to “traditional” information systems analysis and design (ISAD) grammars. This paper has extended their application into the area of process modelling. Results of the analysis in this work have drawn into question the robustness of certain BWV ontological constructs *e.g.*, *conceivable state space*, *lawful state space*, *conceivable event space*, and *lawful event space*. Finally, the implications of the analysis have been articulated as a set of propositions. In further work, it is intended to operationalise these propositions into a set of hypotheses that can be tested in the field against a mature integrated process modelling user base.

The work in this paper exemplifies two concerns that have consistently arisen within the information systems research community over the time that the BWV ontological models have been developed and used. First, the understandability of the constructs within the models has been criticised by various sections of the research community. Wand and Weber [49] originally defined the constructs using a rigorous set-theoretic language. Even though in many subsequent works the researchers attempted to simplify and clarify the explanation of the constructs by defining them using plain English, the criticism of lack of understandability has remained. Second, Wand and Weber [52], Weber and Zhang [55], and Weber [54] concede the difficulty in applying the models to grammars and scripts produced using those modelling grammars. This difficulty stems from the fact that, although the Wand and Weber constructs have clearly defined set-theoretic definitions, the grammars and scripts to which they are applied have at best loose definitions. Consequently, the analysis performed using the BWV models rely to a large extent on the knowledge and experience of the researcher(s) performing the analysis. Such a situation leads to limitations on the results of work in the area.

In an attempt to address these concerns, further work out of this paper will involve the development of a meta model for the BWV representation constructs. The meta language that has been selected for use is an extended form of the Entity-Relationship (E-R) model [41]. Meta models using a similar meta language have proved popular in explaining the constructs of modern modelling techniques, for example, workflow models [37] and object-oriented schemas [38]. Work is well progressed in this direction. As a part of ongoing research work, the entire meta model for the BWV ontological constructs will be compared with existing meta

models for different process modelling grammars. With regard to the level of abstraction, the BWW meta model can be viewed as a meta-meta-model. Relationship meta models between the BWW meta model and the grammar-specific meta models will project the results of the ontological analysis. Situations such as construct excess, redundancy, or overload (*e.g.*, [53]) will be transparent in the models.

Acknowledgement

The authors are indebted to Ron Weber, University of Queensland, Brisbane, Australia, Andreas L. Opdahl, University of Bergen, Norway, and participants at the workshops on ontological issues in information systems at the University of Queensland in 1998 for their helpful comments.

References

- [1] van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains. In: Backhouse, R.C., Baeten, J.C.M. (eds.): Computing Science Rep. 98/01. University of Technology, Eindhoven (1998)
- [2] Bansler, J.P., Bodker, K.: A reappraisal of structured analysis: Design in an organizational context. *ACM Transactions on Information Systems* 11 (1993) 2, 165-193
- [3] Banville, C., Landry, M.: Can the field of MIS be disciplined? *Communications of the ACM* 32 (1989) 1, 48-60
- [4] Batra, D., Hoffer, J.A., Bostrom, R.P.: Comparing representations with relational and EER models. *Communications of the ACM* 33 (1990) 2, 126-139
- [5] Becker, J., Rosemann, M., Schütte, R.: Business-to-business process integration: Functions and methods. In: Galliers, R. *et al.* (eds.): Proceedings of the 5th European Conference on Information Systems (ECIS '97), Vol. II, Cork, Ireland (1997), 816-827
- [6] Benbasat, I., Weber, R.: Research commentary: Rethinking "diversity" in information systems research. *Information Systems Research* 7 (1996) 4, 389-399
- [7] Bunge, M.: *Treatise on Basic Philosophy: Volume 3: Ontology I: The Furniture of the World*, Reidel, Boston (1977)
- [8] Chen, P.P.-S.: Entity-Relationship Model: Towards a Unified View of Data. *ACM Transactions on Database Systems* 1 (1976) 1, 9-36
- [9] Curran, Th., Keller, G.: *SAP R/3 Business Blueprint*. Prentice Hall, New Jersey (1998)
- [10] Curtis, B., Kellner, M.I., Over, J.: Process Modeling. *Communications of the ACM* 35 (1992) 9, 75-90
- [11] Davenport, T.H.: *Process Innovation. Reengineering Work through Information Technology*, Boston (1993)
- [12] Davenport, T.H., Stoddard, D.B.: Reengineering: Business Change of Mythic Proportions? *MIS Quarterly* 18 (1994) 2, 121-127
- [13] Ellison, M., McGrath, G.M.: Recording and analysing business processes: An activity theory based approach. *The Australian Computer Journal* 30 (1998) 4, 146-152
- [14] van Es, R.: *Dynamic Enterprise Innovation*, 3rd ed., Ede (1998)
- [15] Floyd, C.: A comparative evaluation of system development methods. In: *Information Systems Design Methodologies: Improving the Practice*, North-Holland, Amsterdam (1986), 19-37

- [16] Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow-Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases* 3 (1995) 2, 119-153
- [17] Green, P.: Use of Information Systems Analysis and Design (ISAD) Grammars in Combination in Upper CASE Tools – An Ontological Evaluation. In: Siau, K., Wand, Y., Parsons, J. (eds.): *Proceedings of the 2nd CAiSE/IFIP8.1 International Workshop on the Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'97)*, Barcelona, Spain (1997), 1-12
- [18] Hammer, M.: Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review* 68 (1990) 7-8, 104-112
- [19] Hammer, M., Champy, J.: *Reengineering the Corporation. A Manifesto for Business Revolution*, New York (1993)
- [20] Hirschheim, R., Klein, H.K., Lyytinen, K.: *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge University Press, New York (1995)
- [21] Hirschheim, R., Klein, H.K., Lyytinen, K.: Exploring the intellectual structures of information systems development: a social action theoretic analysis. *Accounting, Management and Information Technologies* 6 (1996) 1/2, 1-64
- [22] Iivari, J., Hirschheim, R., Klein, H.K.: A paradigmatic analysis contrasting information systems development approaches and methodologies. *Information Systems Research* 9 (1998) 2, 164-193.
- [23] Jablonski, St., Bussler, Chr.: *Workflow Management*, Thomson, London et al. (1996)
- [24] Karam, G.M. & Casselman, R.S.: A cataloging framework for software development methods. *IEEE Computer*, Feb. (1993), 34-46
- [25] Langner, P., Schneider, C., Wehler, J.: Petri Net Based Certification of Event driven Process Chains. In: Desel, J., Silva, M. (eds.): *Application and Theory of Petri Nets 1998. Lecture Notes in Computer Science*, Vol. 1420. Springer-Verlag, Berlin (1998)
- [26] Leymann, F., Roller, D.: Workflow-based Applications. *IBM Systems Journal* 36 (1997) 1, 102-133
- [27] Loos, P., Allweyer, Th.: Process Orientation and Object-Orientation - An Approach for Integrating UML and Event-Driven Process Chains. *Institut für Wirtschaftsinformatik. Paper 144. Saarbrücken* (1998)
- [28] Meinhardt, St., Popp, K.: Configuring Business Application Systems. In: Bernus, P., Mertins, K., Schmidt, G. (eds.): *Handbook on Architectures of Information Systems*. Springer-Verlag, Berlin (1998), 651-666
- [29] Mylopoulos, J.: Characterizing Information Modeling Techniques, In: Bernus, P., Mertins, K., Schmidt, G. (eds.): *Handbook on Architectures of Information Systems*. Springer-Verlag, Berlin (1998), 17-57
- [30] Olle, T.W., Hagelstein, J., Macdonald, I.G., Rolland, C., Sol, H.G., Van Assche, F.J.M., Verrijn-Stuart, A.A.: *Information Systems Methodologies: A Framework for Understanding*. Addison-Wesley, Wokingham (1991)
- [31] Parsons, J., Wand, Y.: The object paradigm as a representational model. *Working Paper. The University of British Columbia. Vancouver* (1992), 1-22
- [32] Peterson, J.-L.: *Petri Net Theory and the Modeling of System*, Prentice-Hall, NJ (1981)
- [33] Petri, C. A.: *Kommunikation mit Automaten*, Bonn (1962) (in German)
- [34] Reisig, W.: *Petri Nets: an introduction*. Springer-Verlag, Berlin (1985)
- [35] Rosemann, M.: *Komplexitätsmanagement in Prozeßmodellen*. Gabler-Verlag, Wiesbaden (1996) (in German)
- [36] Rosemann, M.: Managing the Complexity of Multiperspective Information Models using the Guidelines of Modelling. In: Fowler, D., Dawson, L. (eds.): *Proceedings of the 3rd Australian Conference on Requirements Engineering (ACRE '98)*. Geelong, Australia (1998), 101-118

- [37] Rosemann, M., zur Mühlen, M.: Evaluation of Workflow Management Systems – a Meta Model Approach. *Australian Journal of Information Systems* 6 (1998) 1, 103-116
- [38] Saeki, M.: Object-Oriented Meta Modelling. In: Papazoglou, M.P. (ed.): *Proceedings of the 14th International Conference on Object Oriented and Entity Relationship Modelling*. Berlin (1995), 250-259
- [39] Scheer, A.-W.: *ARIS–Business Process Frameworks*. 2nd edn. Springer-Verlag, Berlin (1998)
- [40] Scheer, A.-W.: *ARIS – Business Process Modeling*. 2nd edn. Springer-Verlag, Berlin (1998)
- [41] Scheer, A.-W.: *Business Process Engineering*. 3rd edn. Springer-Verlag, Berlin (1998)
- [42] Scherr, A. L.: A new approach to business processes. *IBM Systems Journal* 32 (1993) 1, 80-98
- [43] Seligmann, P.S., Wijers, G.M., Sol, H.G.: Analyzing the structure of I.S. methodologies: An alternative approach. In: Maes, R. (ed.): *Proceedings of 1st Dutch Conference on Information Systems* (1989), 1-28
- [44] Sinha, A.P., Vessey, I.: End-user data modeling: An ontological evaluation of relational and object-oriented schema diagrams. Working Paper. Indiana University. Indiana (1995)
- [45] Verbeek, M.: On Tools & Models. In: van Es, R. (ed.): *Dynamic Enterprise Innovation*. 3rd edn. Ede (1998)
- [46] Wand, Y., Wang, R.: Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM* 39 (1996) 11, 86-95
- [47] Wand, Y., Weber, R.: A model of control and audit procedure change in evolving data processing systems. *The Accounting Review* LXIV(1) (1989), 87-107
- [48] Wand, Y., Weber, R.: An ontological evaluation of systems analysis and design methods. In: Falkenberg, E.D., Lindgreen, P. (eds.): *Information System Concepts: An In-depth Analysis*. North-Holland (1989), 79-107
- [49] Wand, Y., Weber, R.: An ontological model of an information system. *IEEE Transactions on Software Engineering* 16 (1990) 11, 1281-1291
- [50] Wand, Y., Weber, R.: Mario Bunge's Ontology as a formal foundation for information systems concepts. In: Weingartner, P., Dorn, G.J.W. (eds.): *Studies on Mario Bunge's Treatise*. Rodopi, Atlanta (1990), 123-149
- [51] Wand, Y., Weber, R.: A unified model of software and data decomposition. In: DeGross, J., Benbasat, I., DeSanctis, G., Beath, C.M. (eds.): *Proceedings of the 12th International Conference on Information Systems (ICIS '91)* (1991), 101-110
- [52] Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. *Journal of Information Systems* 3 (1993) 4, 217-237
- [53] Wand, Y., Weber, R.: On the deep structure of information systems. *Information Systems Journal* 5 (1995), 203-223.
- [54] Weber, R.: *Ontological Foundations of Information Systems*, Coopers & Lybrand Accounting Research Methodology. Monograph No. 4. Melbourne (1997)
- [55] Weber, R., Zhang, Y.: An analytical evaluation of NIAM's grammar for conceptual schema diagrams. *Information Systems Journal* 6 (1996) 2, 147-170