Association for Information Systems
AIS Electronic Library (AISeL)

ACIS 2001 Proceedings | Australasian (ACIS)

2001

Using Ontological Ideas to Facilitate the Comparison of Requirements Elicitation Methods

Michael N. Johnstone
Edith Cowan University, m.johnstone@ecu.edu.au

Donald C. McDermid
Edith Cowan University, d.mcdermid@ecu.edu.au

Follow this and additional works at: http://aisel.aisnet.org/acis2001

Recommended Citation
Johnstone, Michael N. and McDermid, Donald C., "Using Ontological Ideas to Facilitate the Comparison of Requirements Elicitation Methods" (2001). *ACIS 2001 Proceedings*. 55.
http://aisel.aisnet.org/acis2001/55

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Using Ontological Ideas to Facilitate the Comparison of Requirements Elicitation Methods

Michael N. Johnstone and Donald C. McDermid

School of Computing and Information Science
Edith Cowan University, Perth, Australia
m.johnstone@ecu.edu.au, d.mcdermid@ecu.edu.au

Abstract

There are a plethora of system development methods available to practitioners, all purporting to be the best method. This variety brings with it an element of choice which can be perceived as a problem in itself, quite apart from the issue of developing a system. This paper uses elements of the ontological framework of Bunge, Wand and Weber to critically examine the constructs of several methods used to develop requirements specifications (and, in particular, business process models), notably the Business Rules Diagram (BRD) Method and the Unified Modelling Language (UML).

Keywords

Information Requirements Determination, IS Methodologies, Ontology

INTRODUCTION

There are scores of methods on the market that can be used to develop information systems. Jayaratna (1994), for example, suggests that there are well over 1000 methodologies. Given this variety (a large portion of which may, it is acknowledged, be attributed to marketing propaganda or trivial differences in notation), questions arise as to the effectiveness of different methods and, indeed, how best to compare methods in a way that provides useful information for the practitioner faced with making such a choice. Approaches to comparison that have been attempted include meta-modelling, feature analysis and comparison against an ideal method. This last approach can be extended by considering not the concrete properties or components of an ideal method, but the abstraction of those properties. This approach has the distinct benefit of being able to compare methods with widely differing philosophical perspectives.

According to Wand and Weber (1990), the disciplines of computer science and information systems have languished due to the lack of formality in the definition of basic constructs. This view is supported by ter Hofstede and van der Weide (1992) and it may be that ontology can assist with this issue by providing a formal framework for construct definition. More recent work by van Belle (1996) suggests that this may indeed be so. Ontology is a branch of Philosophy which deals with the real world (Bunge, 1977). It uses theories built with mathematics or logic to reason about the world. The ontology of Bunge has been applied to information systems in several areas, notably the structure of information systems (Wand and Weber, 1995), modelling information systems (Wand and Weber, 1989) and the expressiveness of grammars (Wand and Weber, 1993). Given that Bunge's ontology provides a model and mechanism for abstracting ideas with common properties (in this case, logic), it provides a practical way to compare different development methods.

This paper explains how ontological constructs can be used to compare development methods, briefly describes the two approaches to be compared, highlights some interesting aspects of the comparison, and then analyses the effectiveness of the ontological model. Finally, we make some observations and recommendations for future research.

ONTOLOGY

The idea of using a model (in this case, ontology) to structure thought processes and facilitate reasoning about a system is not new. For example, a well-known metaphor for the internal processing of information is the Mental Model approach of Johnson-Laird (1983, p10) described thus: "*human beings understand the world by constructing working models of it in their minds. Since these models are incomplete, they are simpler than the entities that they represent. In consequence, models contain elements that are merely imitations of reality - there is no working model of how their counterparts in the world operate, but only procedures that mimic their behavior*.". It might be argued that because logic uses non-realistic constructs (predicates, quantifiers etc.) it – and by extension, ontology – has no place in any discussion concerning real-world systems. This is a problem faced by many developers of diagrammatic techniques used for conceptual modelling in that there is an underlying (unproved) assumption that the more realistic the model, the more useful it will be, thus by such reasoning we consider that static diagrams are better representations of the real world than text descriptions and that animated graphics are better than static diagrams etc. This is an example of what Scaife and Rogers (1996) call the "resemblance fallacy".

Wand and Weber (1993) provide a model which is consistent with Johnson-Laird's views above but which extends it in a way directly relevant to the systems development process. They speak of the mathematics of mappings in which the real-world is mapped via scripts onto the machine world (fig. 1). Many scripts are necessary in information systems development because typically there are many intermediate models (design worlds) between the real world and the machine world (program code). In the context of this paper, the mapping between the real world and 'script 1' is the key issue being discussed.

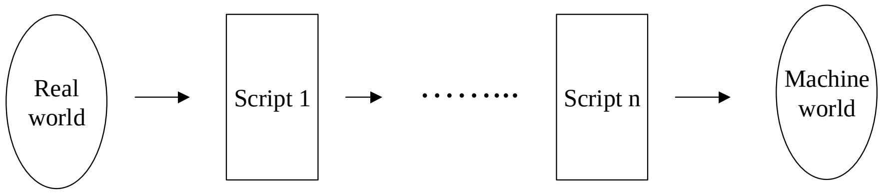

Figure 1: Mapping constructs from the real world to the machine world as a set of scripts, adapted from Wand and Weber (1993).

Wand and Weber (1989) propose that a system can be represented by a tuple of stable states, events and laws. In later work, Weber (1997) describes three basic models, *viz* a representational, a state-tracking and a compositional model which together articulate a more complete ontological system. The model in this paper is not aimed at the complete ontology. The question addressed in this paper is what is are the constructs of other diagrammatic techniques and to what extent do these map onto Wand and Weber's model? The next two sections provide overviews of two approaches.

THE BUSINESS RULES DIAGRAM METHOD

A recent approach used to model system requirements is that of McDermid (1998). This approach, called the Business Rules Diagram (BRD) method, is a state-based model which uses notation similar to flowcharts. As an information systems approach, the BRD method is positioned between the use case approach of Jacobson *et al.* (1992) and more complex object models. A business rule, as defined by McDermid (1998), contains four explicit constructs, these being states, events, conditions and signals. Connected combinations of these constructs make up a User Business Rule Diagram (UBRD) **-** effectively a single business rule. One other construct is the Harel blob (Harel 1988), which encapsulates other constructs and is used to model selection or simultaneous action. The use of the blob construct in the full BRD distinguishes the BRD from the UBRD States reflect the status of a system or one of its components, for example a visitor to a electronic journal web site might traverse the states *visiting, subscribed* and *unsubscribed*. Events are actions carried out internally by the organisation. An important role of the event is to avoid specifying processing detail which is kept separate from higher-level business rules. Conditions define the criteria by which objects of interest in the business move from one state to the next as events take place and are sometimes known as "if-then rules" in other systems. Lastly, signals either enter or leave the human activity system. Signals that enter the system typically initiate activity within the system and are called triggers. Signals which leave the system serve to inform those outside the system boundary about what has occurred inside the system and are called messages. The BRD method also uses a tabular model, called the Event Specification Table (EST) to abstract the structure of the BRD models.

THE UNIFIED MODELLING LANGUAGE

The Unified Modelling Language (UML) is a consistent set of (mostly) graphical models which can be used to assist in the specification and development of information systems. The UML was designed by Booch et al. (1999) to work primarily with object-based modeling but can be used to model data-oriented systems or process-oriented systems. Booch et al. (1999) state four basic principles of modelling which can be paraphrased as: The choice of model affects the solution (cf data vs process modelling); any model may be expressed at various levels of abstraction (complexity); good models are connected to reality (hence the potential usefulness of ontology); and no single model is adequate for a nontrivial system.

The UML has its roots in other methods, notably Rumbaugh's OMT, the Booch method and Jacobsen's OOSE. The current release of the UML is version 1.3 (OMG, 2000), which is the version used in this comparison (a draft of version 1.4 was released in February, 2001).

Given that the focus in this paper is on business process modelling, it is sensible to confine the analysis to those components of the UML that relate directly or indirectly to state-oriented process modelling, namely statechart diagrams (although it is clear that other UML models, such as collaboration, sequence and activity diagrams also model dynamic aspects of systems).

COMPARISON AND ANALYSIS

The ontological constructs (*O*-space constructs) are the anchor points of the framework. If there is a corresponding construct in a method (*A*-space), then there is one-to-one mapping which is denoted "ontological completeness". If there were no corresponding construct, then the method would be considered to be ontologically deficient. Alternatively, if a construct in *A*-space has no ontological analogue (a situation which Wand and Weber describe as "construct excess"), then this may indicate that there has been an element introduced which is outside the scope of *O*-space i.e. the ontological model is lacking in some respect.

O-space constructs		*A*-space constructs			
Ontological construct	Definition*	BRD construct	Definition	UML construct	Definition**
Thing	The elementary unit of the ontological model	×	Comment: equivalent of an object but objects are deliberately hidden in BRD	Element	An atomic constituent of a model
State	The vector of values for all attribute functions of a thing	State	A marker which denotes the status of a system or one of its components	State	A condition or situation during the life of an object during which it satisfies some condition, performs some activity or waits for some event
Event	A change of state of a thing	Event	An action carried out internally by an organisation. Marked by a state to state transition	Event	The specification of a significant occurrence that has a location in time and space; an occurrence that can trigger a transition
Law	A restriction on the values of properties of a thing to a subset considered to be lawful	Condition	The criteria by which things of interest in the business move from one state to the next as events take place	Constraint	An extension to the semantics of a UML element, allowing the addition of new rules or the modification of existing ones
×	Comment: perhaps a signal or message is assumed to be embedded in a law?	Signal	An artifact that either enters (a trigger) or leaves (a message) the system for the purpose of communication with reality external to the system	Message	A specification of a communication between objects that conveys information with the expectation that activity will ensue
Property	Things possess properties. A property is modelled via an attribute function that maps the thing into some value	×	Properties are deliberately hidden at this level of abstraction	Property	A named value denoting a characteristic of an element
Class	A set of things possessing a common property	×	Comment: Since objects are hidden, so are classes	Class	A description of a set of objects that share the same attributes, operations, relationships and semantics

Kind	A set of things possessing two or more common properties	Blob	An artifact which encapsulates either atomic constructs or itself	×	Comment: arguably achieved by inheritance etc.

Table 1: The Ontological Comparison Framework.
* Ontological definitions taken from Wand and Weber (1993) where possible.
** UML definitions taken from OMG (2000) where possible.
× No equivalent in the framework.

As table 1 demonstrates, the ontological elements map quite well onto methods that are inherently state-based such as the BRD and the state-oriented components of the UML. For example, the *O*-space constructs of state, event and law map almost directly into the corresponding *A*-space BRD constructs state, event and condition and the UML constructs state, event and constraint.

However, there are arguably identifiable gaps in table 1 – as it turns out in each of the models. Comments have been inserted in table 1 where a gap has been identified to provide the reader with a possible explanation or even defence. A key issue here concerns not so much the completeness of the approach in terms of coverage of all constructs but rather the wisdom of implying that a single model or diagram should contain all possible constructs given the amount of information that would need to be displayed. Another interesting observation is that the concept of signals poses a problem for Wand and Weber's model. Signals are an example of a class structure in the BRD method but they may be seen as an instance of a class rather than an abstract class. In the UML, internal signals are known as messages. In this framework however, there is no *O*-space construct which can represent either BRD signals or UML messages. This indicates that the ontological model may be deficient.

The framework has concentrated on the atomic constructs of the methods examined. It could be extended to cover complex constructs, for example, the statechart diagram of the UML and the object life history of the BRD are equivalent constructs (related to finite state machines) that are not currently catered for in the ontological model.

This paper has compared the elements of two notations for system development methods against an ontological framework. The work has highlighted a number of interesting issues in requirements elicitation including completeness and information overload in diagrams. More fundamentally, as an insight it has demonstrated a generic approach to evaluating models or diagrams or indeed designing new diagrams. Further work will extend this framework by modifying it to accommodate not just notation but different types of real-world problem situations.

REFERENCES

Booch, G., Rumbaugh, J., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley.

Harel, D. (1988). 'On Visual Formalisms'. *Communications of the ACM, 31*(5), pp514-30.

Jacobson, I., Christerson, M., Jonsson, P., and Övergaard. G. (1992). *Object-Oriented Software Engineering*. Reading, MA: Addison-Wesley.

Jayaratna, N. (1994). *Understanding and Evaluating Methodologies*. London: McGraw-Hill.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge, Massachusetts: Harvard University Press.

McDermid, D. C. (1998). 'The Development of the Business Rules Diagram', PhD thesis, Curtin University of Technology.

OMG (2000). *OMG Unified Modeling Language Specification: Version 1.3*. London: [WWW] available at: http://www.omg.org/technology/documents/formal/uml.htm.

Scaife, M., and Rogers, Y. (1996). 'External Cognition: How do Graphical Representations Work?'. *International Journal of Human-Computer Studies, 45*, pp185-213.

ter Hofstede, A., and van der Weide, T. (1992). 'Formalization of Techniques: Chopping Down the Methodology Jungle'. *Information and Software Technology,* 34*(1)*, pp57-65.

Van Belle, J-P. (1996). 'A Critique of Current System Engineering Methods: The Case for Ontology-augmented Methodologies'. *Section C of*: Siau, K. and Wand, W. (eds), *Proceedings of the Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, Crete, Greece.

Wand, Y., and Weber, R. (1989). 'An Ontological Evaluation of Systems Analysis and Design Methods'. *Pages 79-107 of:* Falkenberg, E. D. and Lindgreen, P., (eds), *Information Systems Concepts: An In-depth Analysis:* North-Holland.

Wand, Y., and Weber, R. (1990). 'An Ontological Model of an Information System'. *IEEE Trans. on Software Engineering*, 16*(11)*, pp1282-1292.

Wand, Y., and Weber, R. (1993). 'On the Ontological Expressiveness of Information Systems Analysis and Design Grammars'. *Journal of Information Systems*, *3*, pp217-37.

Wand, Y., and Weber, R. (1995). 'On the deep structure of information systems.' *Information Systems Journal* 5, pp203-223.

Weber, R. (1997). *Ontological Foundations of Information Systems*, Coopers & Lybrand Accounting Research Methodology Monograph No. 4, Melbourne.

White, B. (1989). 'The Role of Intermediate Abstractions in Understanding Science and Mathematics'. Proceedings of the 11th Annual Meeting of the Cognitive Science Society, pp972-979. Hillsdale, New Jersey: Lawrence Erlbaum Assoc.

COPYRIGHT

Michael N. Johnstone and Donald C. McDermid © 2001. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.