

AN ONTOLOGICAL EVALUATION OF JACKSON'S SYSTEM DEVELOPMENT MODEL

Fiona Rohde

Department of Commerce
The University of Queensland, 4072.
Australia

ABSTRACT

Within the discipline of information systems, numerous methodologies exist to aid the analyst and designer develop a well-designed system that meets the specifications put forward by the user. These methodologies tend to fall into two categories: (a) reality-driven, or (b) information-systems driven. Wand and Weber (1989), have mapped a set of ontologically-based constructs onto two widely-used analysis and design techniques being data flow diagrams (an information-systems driven methodology) and entity relationship modelling (a reality-driven methodology).

In this paper, Wand and Weber's work is extended to investigate the systems analysis and design methodology created by Michael Jackson, namely Jackson System Development (JSD). JSD was selected as it combines the reality-driven and information-system driven approaches by modelling what is happening in reality through the use of constructs related to the process to be modelled (usually computerised processes). The ontological analysis of JSD undertaken during this paper is used to evaluate the methodology's strengths and weaknesses as a modelling technique and ultimately to investigate its usefulness as a systems analysis and design methodology.

INTRODUCTION

An important goal of any system developer should be to produce a system that is well designed. According to Bergland (1981), a well-designed system implies a good system structure. To produce a system that has good structure, methods need to be available to help the designer or analyst achieve their goal. Over the years, a large number of methodologies have become available to assist during the process of analysis, design, and implementation of an information system, Olle (1982). During the same period, numerous studies including Olle (1982) and Floyd (1986), have been conducted in an attempt to gain an understanding of and to aid in an evaluation of the methodologies. This paper will review a framework in which an evaluation can take place. It will then use this framework to evaluate the system analysis and design methodology, Jackson System Development (hereafter referred to as JSD), developed by Jackson in 1983.

To be able to determine the modelling power of JSD, some criteria are needed against which the derived system structure can be evaluated. Wand and Weber (1989, 1990) have proposed an ontological model for an Information System. They "formally define and articulate a number of fundamental information systems concepts", Wand and Weber (1993, p 218). In another work, Wand and Weber (1989) used their ontological framework to examine the

modelling power of two systems analysis and design methodologies: data flow diagrams and entity-relationship modelling.

To perform an evaluation of different methodologies, fundamental information systems concepts underlying the methodologies need to be clarified and defined precisely. By clarifying these underlying concepts, a theoretical base can be developed upon which the different methodologies can be evaluated. The model proposed by Wand and Weber, describes and defines some of the fundamental information concepts. It also develops a theoretical basis that allows the different features of the methodologies to be clarified and evaluated. For these two reasons the Wand and Weber model has been selected as the basis for evaluating JSD.

The format of this paper is as follows. Section Two reviews the ontological model of an information system proposed by Wand and Weber, while Section Three contains a brief review of JSD. Section Four contains an ontological evaluation of JSD, together with an integration of the methodology and model. Section Five summarises the paper and discusses any limitations together with areas for further investigation.

WAND AND WEBER MODEL

As stated previously, it would be advantageous to compare different systems analysis and design models. To make such a comparison, models that are both information systems and reality driven need to be compared. Additionally, each methodology's capabilities to model both the static and dynamic aspects of the modelled domain. By utilising such a basis for comparison, the modelling power of very different systems analysis and design methodologies can be evaluated though the use of the same underlying constructs.

Ontology is one branch of philosophy that deals with both, modelling the existence of things, as well as the structure and behaviour of the world. The Wand and Weber model is based on the "ontological formalism proposed by Bunge (1977, 1979) having been extended and modified for the purpose of modelling information systems" Wand and Weber (1989, p 84).

This section briefly describes a number of ontological (real-world) constructs that form the foundation of the work carried out by Wand and Weber in this area. As JSD is to be evaluated according to this ontology, it is important to provide a brief description of the constructs the aforementioned authors have derived from Bunge's ontology. The terms and concepts defined here are required in Section Four of this article to examine the ontological aspects of JSD. The following definitions are adapted from those found in Wand and Weber (1989, 1990, 1993).

Definition: State Variable: The STATE VARIABLE, v_i , of a thing Y to be modelled is the system identity for the property used to describe that thing in the real world ("thing" is the atomic element within the model and "the world is made of things that have properties" Wand and Weber (1989 p 84)).

Definition: Possible State Space: The Cartesian product of the sets of all possible values of each state variable is the POSSIBLE STATE SPACE.

Definition: Laws: Restrictions (referred to as LAWS) may be placed on either the possible values of the properties of the thing to be modelled or on the relations existing between two or more properties.

Definition: Lawful State Space: The LAWFUL STATE SPACE of a thing to be modelled is the subset of the possible state space where the combinations of state variable values are deemed lawful. This state space, by definition, must be equal to or smaller than the possible state space.

Definition: Event: An EVENT is a change of state. The notion of an event is extended later to include a definition for both external and internal events.

Information systems are usually implemented using data (implementations of states and events) and processes (implementations of laws)

Definition: Transformations: Given the lawful state space for a thing to be modelled, the set of TRANSFORMATIONS from the lawful state space onto itself are the transformations that are given as lawful in the system. An event, as defined above, is considered to be a LAWFUL EVENT if the two states s and s' are both within the lawful state space and the transformation is lawful.

Definition: Coupled: Two things to be modelled, X and Y , are COUPLED if and only if X acts on Y or Y acts on X ; i.e., X changes Y 's history of values or Y changes X 's history of values.

Definition: System: A SYSTEM is a set of things, where each thing is coupled to at least one other thing in the set. Furthermore, the set cannot be partitioned into smaller sets without losing one or more of the couplings. That is, the set of things can be represented by a connected graph.

Definition: Subsystem: To be able to arrive at this definition, three additional concepts need to be defined: the composition, the environment, and the structure of a system. To arrive at a definition for each of these concepts, assume we have a system s .

The composition of s at time t is the set of things modelled in s at time t .

The environment of s at time t is the set of things that are not components of s at time t but act upon or are acted upon by components of s at time t . That is, those things that are not components of s but are coupled to the components of s .

The structure of s at time t is the set of couplings among the components of s and the set of couplings among the things in the environment and the components of s at time t .

Now we can arrive at a suitable definition of a subsystem. x is a SUBSYSTEM of s (denoted $x < s$) if and only if

1. x is a system in its own right at time t , and
2. the components of x at time t are a subset of the components of s at time t , and
3. the structure of x at time t is a subset of the structure of s at time t , and
4. the environment of x at time t is a subset of the combination of the environment of s at time t and those components of s at time t that are not components of x at time t .

Definition: Stable State: Given a state and a set of transformations, the state is said to be STABLE with respect to the set of transformations if and only if the state does not alter unless forced to by an external event.

Definition: Unstable State: Given a state and a set of transformations, the state is said to be UNSTABLE with respect to the set of transformations if and only if the state will be forced to change because of transformations within the system, i.e., internal events.

Definition: External Event: An event is an EXTERNAL EVENT if and only if a change of state occurs because of an interaction with something within the environment.

Definition: Internal Event: An event is an INTERNAL EVENT if and only if the event arises because of lawful transformations.

For a system to act as an information system there are four requirements that are necessary and sufficient for an information system to provide full knowledge of the states that the real system traverses. These four requirements are (a) states of the real system can be mapped to states of the information system, (b) there is a mirroring of external events in the real system to external events in the information system, (c) the information system must interact with its environment, therefore the things in the system must interact with things in the information system and (d) events in the real system will trigger their counterpart in the information system in the same order of occurrence.

Definition: Decomposition: Wand and Weber (1990) define decomposition in terms of the elements a system, that is, the components, the environment, the structure, and the subsystems. A DECOMPOSITION is a set of subsystems of a system where

- (a) every component of the system is included in at least one subsystem;
- (b) the difference between the union of the environments of the subsystems and the components of the system is the environment of the system;
- (c) each element in the structure of the system is included in at least one subsystem of the system.

As stated previously, the ontological constructs defined above, will be used during section four of this paper. The constructs are used to examine to ontological aspects of Jackson's System Development Methodology.

JACKSON'S DESIGN MODEL

There are many different systems analysis and design methodologies. This section describes the System Development Methodology (JSD) that evolved from work carried out by Michael Jackson, Jackson (1983). JSD was selected for two reasons. First, the methodology, while not having widespread use in Australia, does have numerous supporters throughout Europe. Second, JSD is an interesting methodology in that it employs constructs related to the computerised processes as a means of modelling objects in reality. Thus, the methodology can be viewed as a combination of the reality-driven methodologies that focus on constructs related to reality and information-driven methodologies that tend to focus on data and processes.

The methodology comprises six steps or stages. The first four are concerned predominantly with deriving a specification of the required system. The remaining two are concerned with the implementation of the above specification. The major aim of JSD is to ensure the final system is a true reflection of both the developer's and user's perceptions of the real world. The methodology starts by taking the current state of the real world and describing it. By insisting the developer first makes an abstract description of the real world, the methodology is trying to ensure the way in which the user views the real world is incorporated into the specification and eventually part of the system itself.

The real worlds modelled in JSD are always worlds in which the time dimension is of central importance, Jackson (1983, p 16), i.e. dynamic real worlds. When referring to such a world, not only does the analyst take into consideration the items in the real world and the actions they perform or suffer but also the order of the events and the properties and relationships existing among them.

Although there appears to be six independent steps to this methodology as the developer learns more about the system itself, some iterations may be necessary. Since the methodology is not commonly employed throughout Australia, each of the six steps will be briefly described in turn.

Entity Action Step - The developer describes, in abstract terms, part of the real world in terms of entities and the actions in which they participate. General criteria are provided for identifying entities and actions. Jackson (1983, p 40)

Entity Structure Step - The ordering of an entities actions needs to be expressed. To represent entity structure and the time ordering of actions, JSD has introduced the diagrammatic notation known as Structure Diagrams. Actions are applied to an entity as a sequence, as part of an either-or selection, or repetitively (iteration).

Initial Model Step - Thus far the developer has derived an "abstract description of the real world in terms of sequential processes", Jackson (1983, p 44). This step converts the abstract description to a specification of the system by stating how the processes existing in the real world are connected to the processes in the model. The process connection is portrayed using a System Specification Diagram. Additionally, the internal details of the model processes can be specified using structure text.

The Function Step - The developer connects newly-defined functions to the model processes already existing in the System Specification Diagram. The following should be documented as part of this phase: restate the informal requirements for each function in terms of the initial model; expand the System Specification Diagram by illustrating how each function can be connected to a process or may be built into them; and show the detail of each function using structure text.

System Timing Step - Once a specification for a system has been produced the developer can consider the question of timing and how different time lags within processes may affect the output, determining which are acceptable and which are not. The system timing, although usually only informally documented, is one of the major sources of input to the final phase of implementation.

Implementation Step - As soon as the specification derived is considered complete the developer then draws upon the Jackson Program Design techniques to eventually derive the remaining program text.

JSD is an enhancement of Jackson's Structured Programming, Jackson (1975), by taking into account the needs of not only the programmers but also the system developers. JSD's one main advantage over many other methodologies is the introduction of a preliminary stage that concentrates on process modelling and control. JSD uses the preliminary stage of the methodology to investigate the real world and to actually model this world. Once the model of the real world has been derived, specification of the system can be undertaken. This specification is the starting point of the next phase where the data structures are investigated and finally a program written.

ONTOLOGICAL DISCUSSION OF JSD

So far this article has been concerned with describing the Wand and Weber model of a system and the system analysis and design methodology proposed by Jackson. This section seeks to integrate the two by describing Jackson's Methodology using the terminology employed in the Wand and Weber model.

An information system is simply a model of some real-world system. Two steps are involved in the construction of a system: first, create a model of reality; and second, transform this model of reality into the information system. Wand and Weber (1989) state the modelling method chosen can fall into one of two categories: reality driven or information system driven. Jackson's methodology uses processes to model reality. A good methodology, when deriving the decomposition of a system, should be able to represent both the static aspects of a system (structure) and the dynamic aspects of a system (behaviour).

Certain ontological constructs present within the Wand and Weber model need to be examined to determine how these constructs are represented within the methodology. In the remainder of this section, the concepts and definitions given in Section Two of this article, will be used to evaluate the modelling power of Jackson's methodology – that is, JSD is examined in terms of how various ontological constructs are represented. The modelling capacity and thus the strengths and weaknesses of JSD can be assessed by mapping the ontological constructs from the Wand and Weber model onto the constructs developed during the use of JSD. The following ontological constructs will be investigated: things; properties; possible state space; lawful state space; integrity and stability laws; events; coupling; system; environment; and decomposition.

The above constructs represent the set of basic constructs from which any others defined in Section Two can be constructed. For example, once "things" are represented, then "components" can be extracted from the system. Similarly, the "structure of a system" can be determined using the "things," "couplings," and the "environment." Thus the following ontological constructs form the minimal set that need to be discussed.

Things - A thing is the atomic element within the model proposed by Wand and Weber. Jackson's methodology identifies entities and actions during its description of the real world. Using the criteria for identifying entities and actions, both must exist or take place in the real

world outside the system and not merely be a part or an action of the system itself. Entities may be viewed as items that exist in the real world. As such, they can be considered as "things" in the Wand and Weber model. Entities will have certain attributes associated with them. These attributes are represented by state variables. Actions are behaviours that entities either suffer or perform. As such, they represent the dynamic aspects of the system. They will be considered later under the heading events.

Properties - A property is something that is used to describe the "thing" in the real world. Thus, in JSD how are the properties used to describe the JSD entities represented? Properties are not explicitly represented on the Structure Diagrams or System Specification Diagrams. Properties are included in the narrative versions of the processes (process structure text) and functions (Structured English).

State Space and Lawful State Space - Properties are not explicitly represented within JSD, and thus the possible values of the state variables representing these properties are also not represented. The Cartesian products of the set of all possible values of each state variable is the possible state space. Since the possible values for each state variable are not represented, the Cartesian product cannot be determined. Thus the possible state space cannot be represented. The lawful state space is a subset of the possible state space after laws have been considered. Some constraints on values may be reflected through the structured text. Most cannot be represented, however, as there appears to be no appropriate construct. If few constraints on the values are explicitly shown, then the lawful state space cannot be represented.

Integrity and Stability Laws - Wand and Weber (1989) state an integrity law is a constraint on the allowed states of a thing that enables system statics to be modelled. A stability law, on the other hand, is a function that maps a thing from an unstable state to a stable state, thus enabling the dynamics of a thing to be modelled. The processes are defined in detail in the Structure Diagrams. These diagrams, together with the structured text, describe the actions that the entity may perform or suffer.

The actions for each entity tend to be expressed in a procedural manner – that is, the procedures each of the state variables undergo while the action is performed or suffered are represented in the structure text and structured English sections. The actions are not explicitly listed as states and state changes. Therefore the integrity and stability laws are not explicitly represented in JSD in terms of states. They tend to be shown as restrictions on the procedures the state variables may undertake.

Events - An event is a change of state that arises because the value of at least one state variable alters. There are two types of events: internal and external. An external event is where a state variable representing a property of a thing alters because of some action of a thing in the environment. An internal event is where a state variable alters because some action is performed or suffered by a thing not in the environment. Events are represented by the actions that entities may perform or suffer. These are represented by the actions on the structure diagrams. We need to determine whether both internal and external events are represented. JSD allows for processes to exist in either the real world or within the system. Those within the system (postscripted –1 or greater) may perform actions that are internal events. Those postscripted –0 represent processes within the real world or the environment of the system. These actions impact on a thing in the system and thus are external events.

After examining the manner in which the preliminary stage of JSD describes the specification of the real world by concentrating on process modelling and control the following requirements of information system should be properly considered: (a) states of the real system should be mapped to states of the information system, (b) there should be a mirroring of external events in the real system to external events in the information system, (c) the information system's interaction with the environment must be considered and (d) events in the real system should trigger their counterpart in the information system in the same order of occurrence.

Coupling - Coupling between two things is the ability of one thing to affect the history of another thing. Within JSD, two types of diagrams are used. Structure charts illustrate the actions that are performed or suffered by entities. System Specification Diagrams illustrate the relationships that exist between processes. These two types of diagrams are interrelated. Structure charts, by illustrating entities and their associated actions are, in fact, a representation of a process. This process is the item depicted within the System Specification Diagram. The actions that an entity performs or suffers may alter the state variables of that entity. When an entity has state variables that alter, it may affect state variables contained within other entities. This alteration of state variables of other entities is due to the state vector and data flow connections. The state variable and data flow connections are the links between processes on the System Specification Diagram. These links between processes thereby represent the couplings. These couplings reflect the fact that some process needs the information contained within another process to work correctly.

Environment - A JSD System Specification Diagram shows which processes are parts of the system and which belong to the environment. As stated previously, those processes postscripted -0 belong to the environment, and those postscripted -1 or greater belong to the system.

System - All processes are represented on a System Specification diagram. So are all the connections between these processes. As stated under the heading "couplings," the links or connections between these processes represent the couplings between components. All links between processes are represented. The question that needs to be raised is "Are these links on the Software Specification Diagram all the couplings?" To answer this question, we need to consider the ways in which state variables can alter and determine how each is represented. First, by one process affecting another process, the coupling is represented by a state vector or data flow connection. Second, the state variable may be altered by actions within process. These actions do not affect any entities outside this process's structure chart and therefore these changes do not indicate couplings. Third, the links to the environment are depicted on the System Specification Diagram. Therefore, the couplings between the environment and the system are depicted. There is no other way by which a state variable can alter and thereby all the couplings between things are represented. Since all couplings have been represented, the system is represented in full by the System Specification Diagram.

Decomposition - JSD tries to model reality directly. Thus, hopefully the structure of the system is a good representation of the events in the real world. However, Floyd (1986) states there is no mechanism for forming abstractions, subsystems, or layers, thereby making it difficult to identify or understand which functions belong together. JSD appears to be able to

capture things in the real system in a time-ordered manner. It allows the user to build up a picture of the system as a whole. However, it lacks a mechanism for saying exactly how the system should be broken up into subsystems. Therefore, it can be viewed as showing an overall decomposition of the system. The analyst must then apply heuristics to decide which processes and functions to place in one subsystem and which to place in another.

The above section has examined the modelling power of JSD in terms of the ontological constructs proposed by Wand and Weber. The manner in which the various ontological constructs are represented within JSD were explicitly examined.

CONCLUSIONS

This article has investigated the ontological aspects of the systems analysis and design methodology proposed by Jackson. The aim of the ontological evaluation was to evaluate JSD in terms of the Wand and Weber model so that the strengths and weaknesses of the methodology can be examined. The article commenced by examining the formal model of an information system proposed by Wand and Weber (1990). The system analysis and design methodology, JSD, was then described in detail before the methodology and model were integrated.

JSD seems to be able to include through the use of its methodology "things" that exist in the real world. These "things" are identified as part of the first step of the methodology and thus the system can be validated against the user requirements at a very early stage in development. This is certainly an advantage as a system that does not comply with the user's perception of reality can be costly to correct.

Many of the other ontological constructs identified by the Wand and Weber model are also explicitly represented by the JSD approach through the use of the diagrammatic notation used. The use of structured text is also an aid to include other ontological constructs not explicitly included in the diagrams.

A good methodology should be able to represent both the structure of the system and also the behaviour of the system. JSD through the use of structure diagrams appear to capture the things in the real world in an ordered manner. Through the aid of these diagrams, the structure of the system as a whole is built up by the designer. The methodology, through the aid of these same diagrams used in conjunction with the structure text, describes the actions the things may perform or suffer. In this manner, some of the behaviour of the system, i.e., the dynamics, are also represented.

During the ontological evaluation of JSD, it was discovered that JSD derives the overall structure of the system to be implemented. How to decompose this structure further into subsystems is not addressed within the methodology. There are no steps to say which entities/processes/functions should be joined to form a subsystem. The decomposition procedure is entirely up to the analyst and is based on heuristic methods.

One major limitation of this study comes from its heavy reliance on the ontological model proposed by Wand and Weber. The model of an information system proposed by Wand and Weber has been taken as correct. From this model, the definitions were derived that were

thought to be necessary to allow JSD to be evaluated using ontological constructs. Any extension of the underlying ontological constructs discussed during this paper may result in an extension to this piece of research.

There are many opportunities for future research in this area. The issues raised during the mapping of the ontological constructs from the Wand and Weber model onto the constructs created during the use of JSD form the basis of the implications and future research directions of this study.

The study could be replicated for other methodologies apart from JSD and those previously investigated by Wand and Weber. Currently there is no other theoretical foundation upon which to systematically evaluate the various methodologies available to the analyst and designer. Through this type of research, hopefully a better understanding of the strengths and weaknesses of each methodology can be evaluated and eventually different methodologies can be compared using the same theoretical base.

Another future research direction that could be pursued relates to extensions of the Wand and Weber model. The aforementioned researchers have extended their model to include different criteria that they consider necessary for a decomposition to be considered a "good" decomposition. The ontological evaluation of JSD could be extended to evaluate different decompositions derived through the use of JSD to determine whether each is or is not a good decomposition. By examining the different decompositions that are obtained by the system analyst and designer and by determining whether or not they are good decompositions, formal steps may be able to be incorporated into JSD to allow the decomposition phase to be achieved with the aid of fewer heuristics.

This paper has contributed to the area of information systems by attempting to provide an ontological evaluation of JSD and thus evaluating the strengths and weaknesses of the underlying methodology. The results indicate that JSD gives the user an overall system structure comprising things described by the minimal number of state variables. However, no formal way of arriving at a decomposition of the system is provided. Considering this limitation of JSD, the overall ability of the methodology as a systems analysis and design methodology was considered, and it was found that many ontological constructs appear to be included.

The results of this study could be added to the previous work undertaken by Wand and Weber, thus providing an examination of an additional systems analysis and design methodology in terms of ontological constructs. As stated in the introduction of this paper, for a proper evaluation of all methodologies, they need to be examined on common ground. This study has now placed as additional methodology on this common ground. The work undertaken here could be extended by now comparing JSD with the other methodologies already examined in terms of their ontological constructs.

REFERENCES

- Bantleman J P And Jones A H (1984) "Systems Analysis Methodologies - A Research Project" in **Beyond Productivity: Information Systems Development for Organisational Effectiveness** edited by Th. M. A. Bemelmans, Amsterdam, The Netherlands: North-Holland
- Bergland G. D. (1981) "A Guided Tour of Program Design Methodologies", **Computer** Vol 14, no. 10, pp 13-37
- Bubenko Jr J A (1986) "Information System Methodologies - A Research View" in **Information Systems Design Methodologies: Improving the Practice**, (OLLE T W SOL H G and VERRIJN-STUART A A, Eds.), Amsterdam The Netherlands North-Holland
- Bunge M (1977) **Treatise on Basic Philosophy: Vol. 3: Ontology I: The Furniture of the World**, Boston: Reidel. .
- Bunge M (1977) **Treatise on Basic Philosophy: Vol. 4: Ontology II: A World of Systems**. Boston: Reidel.
- Courtois P J (1985) "On Time and Space Decomposition of Complex Structures." **Communications of the ACM**, Vol 2, no 6, pp 590-603.
- Floyd C (1986) "A Comparative Evaluation of System Development Methods" in **Information Systems Design Methodologies: Improving the Practice**. (OLLE T W SOL H G and VERRIJN-STUART A A, Eds.), Amsterdam The Netherlands North-Holland
- Jackson M A (1975) **Principles of Program Design.**, London: Academic Press.
- Jackson M A (1983) **System Development.**, London: Prentice-Hall International.
- Martin J And McClure C (1988) **Structured Techniques: The Basis for CASE**. Revised Edition, Englewood Cliffs, New Jersey: Prentice-Hall.
- McNeile A T (1986) "Jackson System Development (JSD)", in **Information Systems Design Methodologies: Improving the Practice**. (OLLE T W SOL H G and VERRIJN-STUART A A, Eds.), Amsterdam The Netherlands North-Holland
- Olle T W (1982) "Comparative Review of Information Systems Design Methodologies" in **Information Systems Design Methodologies: A Compartative Review**. (OLLE T W SOL H G and VERRIJN-STUART A A, Eds.), Amsterdam The Netherlands North-Holland
- Paulson J D, (1988) "Reasoning Tools to Support System Analysis and Design", Ph.D. Thesis, Univ. British Columbia, Vancouver, Canada.
- Pressman R S (1987) **Software Engineering: A Practitioner's Approach**. Second Edition, New York: McGraw Hill.
- Wand Y And Weber R (1989) "An Ontological Evaluation of Systems Analysis and Design Methods" in **Information Systems Concepts - An In-Depth Analysis**, (FALKENBERG E and LINDGREEN P, Eds.), Amsterdam, The Netherlands: North-Holland, pp 79-101.
- Wand Y And Weber R (1990), "An Ontological Model of an Information System". **IEEE Transactions of Software Engineering**, Vol 16, No 11, pp 1282-1292.
- Wand Y And Weber R (1991) "A Unified Model of Software and Data Decomposition." **Proceedings of the Twelfth International Conference on Information Systems**, New York, New York, pp 101-110.
- Wand Y And Weber R (1993) "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars". **Journal of Information Systems**, pp 217-237.