REFINEMENT OF RESEARCH SURVEYING IN SOFTWARE METHODOLOGIES BY ANALOGY: FINDING YOUR PATCH

Eugene Eric Doroshenko
Department of Computer Science and Software Engineering
University of Melbourne
email: eugene@cs.mu.oz.au

Felicity Lear
School of Information Systems
University of Tasmania
email: Felicity.Lear@infosys.utas.edu.au

ABSTRACT

To enhance research surveying in software methodologies, a model is introduced that can indicate field maturity based on vocabulary and relevant literature. This model is developed by drawing analogies with software methodologies. Two analogies are used: software models and software life cycles or processes. How this model can reduce research surveying problems for researchers is described using extracts from application results as examples. Although the model does support research surveying activities, it cannot choose the subject for the researcher.

INTRODUCTION

Part of research in software methodologies is reviewing of literature and identification of candidate projects based on the literature review. The literature review process assists knowledge acquisition by the researcher about a subject. Two foundations of candidate projects are the maturity of the field and unresolved issues in the literature. Areas that are immature are likely to have more candidate projects. The more unresolved issues there are, the more projects that can be derived from them. Hence, the literature review is a foundation for knowledge acquisition as well as project identification based on field maturity and unresolved issues. All of these are part of research surveying activities.

Researchers in software methodologies still face some problems. Firstly, a researcher may have little experience in research surveying; honours, masters, and PhD students are three examples. Secondly, beyond the scope of keywords, number of literature sources, and types of literature sources, much surveying tends to be ad hoc in nature and highly dependent on the experience of the researcher. Thirdly, knowledge acquisition cannot be avoided, but tracking and organising such knowledge for a literature review can be difficult, particularly in mature fields. Some frameworks and models (FMs) classify software methodologies for various purposes, such as research surveying of CASE technology (Dart, et al., 1987; Fuggetta, 1993), research surveying of modelling methods (Doroshenko, 1994; Iivari, 1995; Monarchi and Puhr, 1992; Roper, 1995), evaluation of software process modelling (McChesney, 1995), and technology transfer (Korson and Vaishnavi, 1992). Although the FMs assist knowledge acquisition and research surveying of a subject, they tend to be specific to their subject and lack sufficient generalisation for application in other subjects. In any case, research surveying activities are not dealt with explicitly.

One FM[1] has shown some success in more generalised application. Hence, to deal with the above issues, a model for research surveying in software methodologies is presented using Doroshenko (1994) as a starting point. A number of features are introduced based on FMs that classify software methodologies. A model for construction of research surveying tools that incorporates these features is outlined. The sum of these components are analogous to a software model. A development process is described to show how the model components support research surveying activities. The development process is analogous to a software life cycle. Stages in the development process are literature acquisition, component formation, data gathering, and data analysis. Literature acquisition is done as preparation for component formation. Component formation and data gathering assist knowledge acquisition in a subject. The results of data gathering are used in data analysis to support identification of candidate projects via field maturity and unresolved issues. In the same way that software applications for a problem are built using software models, a research surveying tool for a subject is made using

[1] Doroshenko (1994) described a framework for classifying software methodologies using object-oriented methods. This was applied to structured process and structured data methods by Roper (1995). Both Doroshenko and Roper received first class honours, and Doroshenko went on to further research obtaining a University of Tasmania Scholarship with APA conditions and stipend.

the model components. For supervisors, this work can be presented to research students to assist their research surveying activities and lessen supervisor workload. Limitations and Issues relating to alternative application and specification conclude this paper.

MODEL FOUNDATIONS

Table 1 summarises the foundations for a research surveying tool in software methodologies. Note well, it is the features themselves that are important, not the examples of the features. For example, "dimension" as a feature, not the example of a dimension, such as "aspect dimension" or "viewpoint dimension" in Iivari (1995).

Table 1: Feature Identification based on Sources

Source	Location	Features
(Beringer, 1994)	1-3, 5-7	Dimension, Subdivision, Vocabulary, Concept/Representation
(Curtis, et al., 1992)	76 - 80	Dimension, Subdivision
(Doroshenko, 1994)	19 - 37	Subdivision, Vocabulary, Concept/Representation, Dependency
(Doroshenko, 1996)	162-168	Dimension, Subdivision, Concept/Representation, Dependency
(Fernstrom, et al., 1992)	38	Aggregate dependency
(Fuggetta, 1993)	28 - 38	Subdivision, Vocabulary, Aggregate dependency, Classification dependency
(Henderson-Sellers and Edwards, 1990)	143 - 145	Aggregate dependency, Subdivision
(Iivari, 1995)	156 - 158	Dimension, Subdivision, Vocabulary, Prerequisite dependency
(Korson and Vaishnavi, 1992)	106 - 109	Subdivision, Classification dependency, Vocabulary
(OMG, 1996)	5-7, 14-16	Subdivision, Vocabulary
(Lowry, et al., 1994)	223 - 224	Dimension, Subdivision
(McChesney, 1995)	364 - 368	Classification dependency
(Monarchi and Puhr, 1992)	36 - 40	Subdivision, Prerequisite/Classification dependency, Concept/Representation
(Snyder, 1993)	38 - 42	Concept/ Representation

Location refers to page numbers. In some cases the feature is discussed (e.g. Beringer (1994) for concept/representation), in others it is only evident (e.g. Monarchi and Puhr (1992) for dependency)

Dimensions divide an FM into mutually exclusive areas. e.g. Lowry et al (1994) have methodology, organisation and linkage dimensions. Iivari (1995), based on De Champeaux and Faure (1992), has aspect and viewpoint dimensions. These are usually illustrated as lines at right angles with words (e.g. Figure 1; *Stages, Components*)

Subdivisions divide a dimension into smaller parts. e.g. Beringer (1994) has a model level dimension with analysis and design subdivisions. Iivari (1995) has structure, function, and behaviour subdivisions for the aspect dimension, along with individual and object community subdivisions for the viewpoint dimension. These are usually illustrated as intervals along the lines representing dimensions (e.g. Figure 1; *Analysis, Design, Modelling, Method Process).*

Vocabulary form the basic units for subdivision intersections. e.g. Iivari (1995) has vocabulary for Individual structure (Object, attribute) and Object community structure (Inheritance, subsystem). Monarchi and Puhr (1992) have a process subdivision with identify, placement and specification of classes. These are usually represented as words enclosed by areas delineated by subdivisions (e.g. Figure 1; *Class, Object, Operation, Aggregation, Identify Class, Identify Object, Identify Aggregation*). Korson and Vaishnavi (1992) enumerate the notion of vocabulary description. A conceptual vocabulary item has a "Name:" as the conceptual term, a "type" analogous to its subdivision, source contributions identified as "References", and information defining the term in "Description", "Notation:", and "Example".

An FM can have a division between concepts and their representation. e.g. Snyder (1993) indicates division between concept meaning (Object) and its varying terms (Instance, class instance, object). Doroshenko (1994) has a conceptual model of the method process, and a representation model of modelling notation. Monarchi and Puhr (1992) associate modelling concepts with their graphic representation.

Dependencies can further organise FM parts. These include include:

> **Classification; one FM part is a more general form of another FM part.** e.g. McChesney (1995) has classification dependency for specific concepts, such as style (AI, Petri net, functional, OO) and notation (text or graphic).
>
> **Aggregation; one FM part is composed of, or part of, another FM part.** e.g. Henderson-Sellers and Edwards (1990) have subdivision aggregate dependency, where a phase (Build) is composed of more specific phases (Coding, Program testing, Program use).
>
> **Prerequisite or Existence; one FM part is dependent on another FM part for its own definition.** e.g. Monarchi and Puhr (1992) have the process and representation subdivisions dependent on object-oriented modelling concepts. Doroshenko (1994) has the notation and method process for object-oriented methods dependent on their modelling concepts.

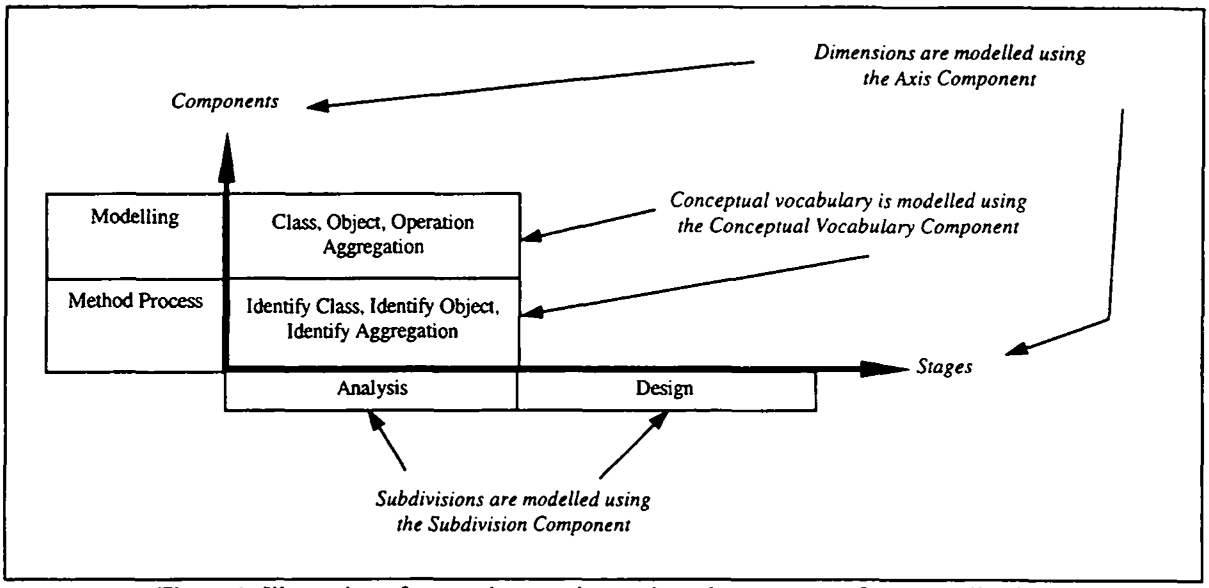


Figure 1: Illustration of research surveying tool, analogous to a software application

The above features appear across many FMs. But each FM usually contains only a subset of the features described, and lacks generalisation for research surveying in other subjects. One FM does indicate a means of tool generalisation using self- definition (Doroshenko, 1996), but it is specific to CASE technology. Doroshenko (1994) is more applicable to research surveying and suggests how some features can be integrated, but lacks sufficient generalisation for other subjects. To integrate the above features into a model for construction of research surveying tools, the author uses Doroshenko (1996) as a foundation for generalisation, and Doroshenko (1994) as a foundation for structure.

Based on the approach in Doroshenko (1996), the model is described using three tools, resulting in a number of model components. The model components extend and generalise the framework in Doroshenko (1994) with the features summarised in Table 1. The specification tools are table templates (Appendix B), ER modelling (Figure 2), and an EBNF variation called the text definition language (TDL, Appendix A) based on four sources ((Backus, 1959; Doroshenko, 1996; Naur, 1963; Sethi, 1989), See Appendix D for an overview of TDL). The ER model gives an overall picture of the model. The TDL and table template specifications are notation alternatives to forming a research surveying tool.

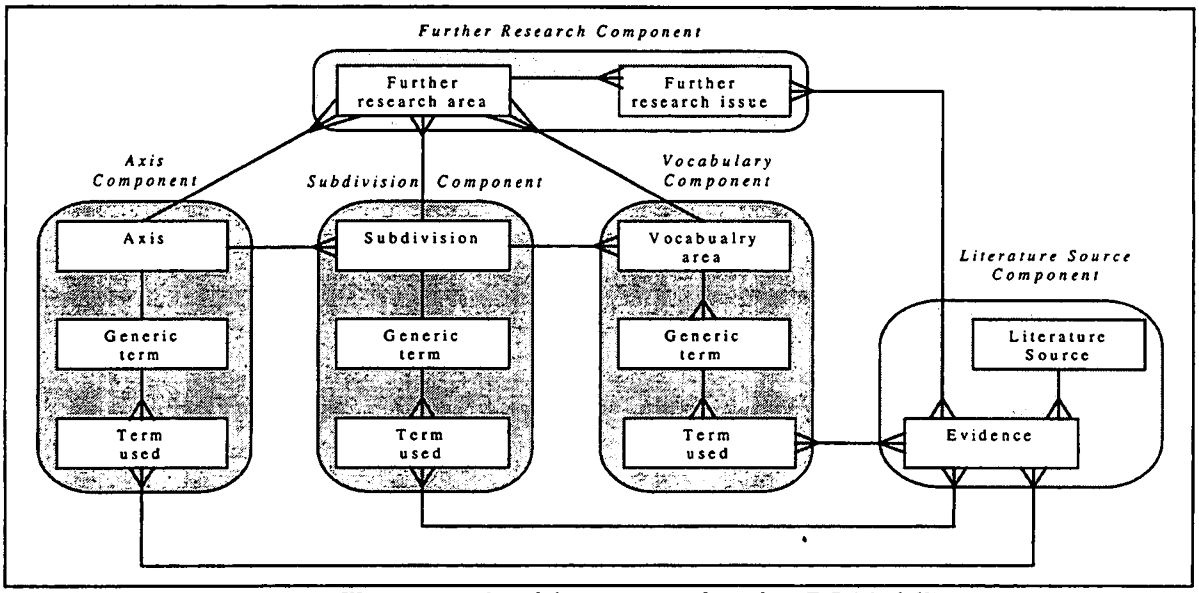


Figure 2 *Illustration of model components based on E-R Modelling*

MODEL COMPONENTS AS A SOFTWARE MODEL

a)

Axis Generic term: Stage
[**Description:** A software methodology is divided into a number of phases that represent periods in time, with each phase setting some aim or purpose for work done in the phase based on the phase name]

Term used	Source	Evidence
Phase	Vlasbolm et al 1996; Henderson-Sellers and Edwards 1990;	page 597, Figure 1a, page 597; page 144;
Stage	Jones 1990;	"...The normal stages of the software life cycle are shown in Figure 1-8. Each product passes through these stages although the duration... of each stage may vary..." (p 11), pages 11- 19
Activity	Fuggetta 1993;	page 28, page 32

b)

Axis Generic term: Component
[**Description:** A software methodology is divided into a number of components that describe its processes and products.]

Term used	Source	Evidence
?	ter Hofstede et al 1997	pages 404 - 405

c)

Stage **Axis.**
Dependencies: Type: Component. Design;
Subdivision Generic term: Logical design
[**Description:** The logical design stage is where a system is developed according to how it will solve a problem, independent of a programming language and hardware environment]

Term used	Source	Evidence
Logical design	Firesmith 1993; Monarchi and Puhr 1992;	pages 15 - 16; page 36 - 37

Figure 3: Example Axis and Subdivision Component Instances

The *axis component* supports formation of axes. Each axis has a *name*, indicating the conceptual name of the axis, and some *interpretation*. For example, in Figure 3a) the name of the axis is "Stage", its interpretation starts as "Description: A Software methodology is divided into a number of phases that represent periods in time...".

An axis has a number of *representation terms* based on *evidence* in *literature sources*. For example, in Figure 3a) the representation terms are "Phase", "Stage", and "Activity". A literature source for the "Activity" representation term is "Fuggetta 1993". The evidence in "Fuggetta 1993" is "page 28, page 32". Figure 3a) and b) illustrate examples of the axis component using its table template specification. See, Table 6 for details on the relationship between the axis component and its specifications in Appendix A and Appendix B.

a)

Analysis Modelling **Conceptual Vocabulary**
Dependencies:
Analysis Modelling **Concetual Vocabulary Type:** Aggregate; Object.
Vocabulary Generic term: Operation
[**Description:** An operation can either access or change the state of an object.]

Term used	Source	Evidence
Operation	Booch 1994	page 43
Service	Henderson-Sellers and Edwards 1994	page 51
Behavior	Monarchi and Phur 1992	pages 39 - 40

b)

Analysis Modelling **Conceptual Vocabulary**
Dependencies Analysis Modelling **Conceptual Vocabulary Type:** Classification. Aggregation ;
Vocabulary Generic term: Owns Aggregation
[**Description:** A class is composed of classes, such that deletion of the class implies deletion of its parts.]

Term used	Source	Evidence
Owns Aggregation	Maciaczek et al 1996;	page 435, paragraph 7, Starter quote: "...The Owns Aggregation..."
Existence Dependent	Rahayu et al 1996;	pagees 524 - 525, Figure 2, page 524

c)

Analysis Method Process **Conceptual Vocabulary**
Dependencies: Analysis Modelling **Conceptual Vocabulary Type:** Prerequisite. Class;
Method Process **Conceptual Vocabulary Type:** Prerequisite Identify
Vocabulary Generic term: Identify Class
[**Description:** An activity that identifies classes in an object model]

Term used	Source	Evidence
Finding classes	Monarchi and Puhr 1992;	"...an important process in OOAD is the identification of classes..." page 39, Table 3 page 39.

Figure 4: Example Conceptual Vocabulary Component Instances

The *subdivision component* supports partitioning of axes based some axis component. An *axis* is *linked* to one or more *subdivisions* based on the axis name. For example, in Figure 3c) "Stage Axis" links the "Stage" axis to the Logical design subdivision. Each subdivision has a *name*, indicating the conceptual term for the subdivision, and some *interpretation*. For example, in Figure 3c) the name for the subdivision is "Logical design". The interpretation for the Logical design subdivision starts as "Description: The logical design stage is where a system is developed according to how it will solve a problem...". A subdivision has one or more *representation terms* based on *evidence* in *literature sources*. For example, in Figure 3c) the logical design subdivision has the representation term "Logical design". Two literature sources for this term are "Monarchi and Puhr 1992" and "Firesmith 1993". The evidence in "Monarchi and Puhr 1992" is "page 36 - 37". Subdivisions can have *dependency* on other subdivisions from the same axis. Dependency types include aggregate, prerequisite or existence, and classification described earlier. For example, Figure 3c) the Logical design subdivision is a part of the Design subdivision (Aggregate dependency). This is indicated by "Type: Component. Design;". Figure 3c) illustrates an example of the subdivision component using its table template specification.

See Appendix C Table 5 for details on the relationship between the subdivision component and its specifications in Appendix A and Appendix B.

The *conceptual vocabulary component* **supports grouping of related vocabulary concepts** based on a *vocabulary area.* The vocabulary area is formed using subdivision names from subdivisions. For example, in Figure 4a) the vocabulary area is "Analysis Modelling Conceptual Vocabulary". This means there are two subdivisions named "Analysis" and Modelling". A vocabulary area has a number of *vocabulary concepts.* Each concept is given a *name,* indicating its conceptual term, and some *interpretation.* For example, in Figure 4a) Operation is a vocabulary concept, its name is "Operation"! Its interpretation is "Description: An operation can either access or change the state of an object.". A concept has one or more *representation terms* based on *evidence* in *literature sources.* For example, in Figure 4a) the vocabulary concept Operation has "Operation", "Service", and "Behavior" representation terms. The literature source for the "Operation" representation term is "Booch 1994". The evidence for this in "Booch 1994" is "page 43". Concepts can have *dependency* on other concepts in vocabulary areas. Dependency types include aggregate, prerequisite or existence, and classification. For example in Figure 4a) the vocabulary concept "Operation" is part of the concept "Object" (Aggregate dependency). This is indicated by "Analysis Modelling Conceptual Vocabulary Type: Aggregate.Object;". In Figure 4b) the "Owns Aggregation" concept is a more specific kind of "Aggregation" concept (Classification dependency). This is indicated by "Analysis Modelling Conceptual Vocabulary Type: Classification.Aggregation;". In Figure 4c) the "Identify Class" concept depends on the "Identify" concept (Prerequisite dependency). This is indicated by "Method Process Conceptual Vocabulary Type: Prerequisite. Identify;". Figure 4a), b), and c) illustrate examples of the conceptual vocabulary component using its table template specification. See Appendix C for details on the relationship between the conceptual vocabulary component and its specifications in Appendix A and Appendix B.

The *further research component* **supports identification of research issues** based on axes, subdivisions, vocabulary areas, and conceptual vocabulary. A *further research component* is *linked* to an *axis, subdivision, vocabulary area, or vocabulary concept(s).* For example, in Figure 5a) the "Stages Activities Phases Further research" component is linked to the "Stages" and "Phases" axes, and the "Activities" subdivision. One or more *research issues* are indicated, along with *evidence* from *literature sources* that contribute to issue identification. For example, in Figure 5a) the description of an issue starts as "Notes: Some authors use phase and stage interchangeably...". A literature source that contributes to issue identification is "Vlasbolm et al 1995". Evidence for the issue in "Vlasbolm et al 1995" includes "Figure 5 page 599". Figure 5a) gives an example of the further research component using its table template specification. See Appendix C Table 10 for details on the relationship between the further research component and its specifications in Appendix A and Appendix B.

The *literature source component* **specifies the various sources of information.** Instances of the literature source component specify the literature in a subject for reference by instances of other model components. (Usually <Evidence entry> or <Literature source>/<Evidence statement> appear as part of the specification). A *literature source* has a unique *identifier* using some referencing system. A *literature source* is *linked* to an *author* or *authors.* For example, in Figure 5b) the identifier for the literature source in "Booch 1994". The author of the literature source "Booch 1994;" is "Author: Booch, Grady;". This is indicated by "Author: Booch, Grady;" following its identifier. Literature sources can be of a particular *type.* Literature sources are *linked* to the *evidence extracts* they contain. For example, in Figure 5b) the type of the source is a book, indicated as "Type: Book;". Two evidence extracts are "p 88." and "Figure 5-24 (p. 209). A number of sources can be included as a literature source component. Korson and Vaishnavi (1992) Figure 2, page 103) illustrate a number of examples. Figure 5b) illustrates an example of the literature source component using its TDL specification. See Appendix C Table 12 for details on the relationship between the literature source component and its specification in Appendix A. Specification of instances can be minimised by using a citation formatting package like Endnote®. The key identifiers for literature sources in Endnote® can be used as instances of <Literature source> in instances of other model components. See the second row of the table in Figure 6a) for an example. Evidence extracts linked to a literature source still need consideration. Specifically, certain terminal and non-terminal sequences in the <Literature source entry> production can be left out. These are "**Authors:** { <Author entry> ; }", <Source title>, <Source type>, <Year published>, and <Publisher>. The "<Author declaration>"can also be left out. But the terminal/non-terminal sequences "**Evidence:** { <Evidence statement> } ;" and "<Literature source> ;" cannot be left out.

a)

Stages Activities Phases **Further research**
Notes: Some authors use phase and stage interchangeably. Fuggetta (1993) shows explicit discrimination between phase and activity, with activities composed of tasks. In contract, Vlasbolm et al (1995) have phases composed of activities. Hence, use of Fuggetta for assessing CASE tools for methodologies formed using Vlasbolm (1995) could be difficult.

Source	Evidence
Fuggetta 1993	"...Tasks are formed to group activities... The activity concept is not to be confused with the phases of a waterfall life cycle. Activities are not necessarily carried out in strict sequence: They can be composed to form any type of life cycle..." (page 28), "...Products for Analysis and Design Activities..." (page 32)
Vlasbolm et al 1995	"...the most striking feature of many methodologies is their way of dividing up the development into phases... development phases are subdivided into activities..." (page 597), Figure 5 page 599
Monarchi and Puhr 1992	"...While identifying clases, attributes and behavior has been fairly throughly covered, placing them has been largely negelected..." (p 41)
Henderson-Sellers and Edwards 1990	"...The design stage is perhaps the most loosely defined since it is a phase..." (page 144), "...Undertake system requirements specification. This stage..." (page 149)

b)

Author: Booch, Grady ;

Booch 1994; **Author:** Booch, Grady; **Title:** Object-oriented Analysis and Design with Applications;
Type: Book; City: Redwood City; **Edition:** 2nd; **Number of Pages:**589;
Year: 1994;
Publisher: Addison-Wesley;
Evidence:
p 88.
Figure 5-24 (p. 209).
"...Figure 5-24 shows the Icon we use to represent an object in an object diagram..." (p. 208).

Figure 5: Example Further research and Literature Source Component Instances

Figure 6a) elucidates application of the table template in Appendix B for the conceptual vocabulary component, resulting in an instance of the conceptual vocabulary component. Figure 6b) illustrates the same instance using the TDL specification in Appendix A. Note the relationship between Figure 6a) & b) and the Vocabulary row in Table 2 and Table 3. Both the table template and TDL specifications use common constructs with a different style. Note also the use of type and instance. For example, using Figure 6a), the name of a concept is specified using "**Vocabulary** <Generic term>", with an instance being "**Vocabulary Generic term:** Identify Class".

ANALOGICAL VIEW OF MODEL COMPONENTS

When using the component specifications, model components can be viewed as as a software model, with the TDL and table template specifications being two forms of notation (Author's preference). Alternatively, the TDL specifications can be viewed as a programming language, with the specifications as record types and the instances as record variables; or the table templates as tables in a database that are filled out with data. Table 2 summarises the relationship between the features in Table 1 and TDL specifications for model components in Appendix A Table 3 summarises the relationship between the features in Table 1 and Table Templates for model components in Appendix B. The specifications usually reflect the descriptions of the components above. For example, the axis component has an axis *name*, which is specified using the non-terminal <Generic term> in <Axis name>. Further details about model component specification can be found in Appendix C.

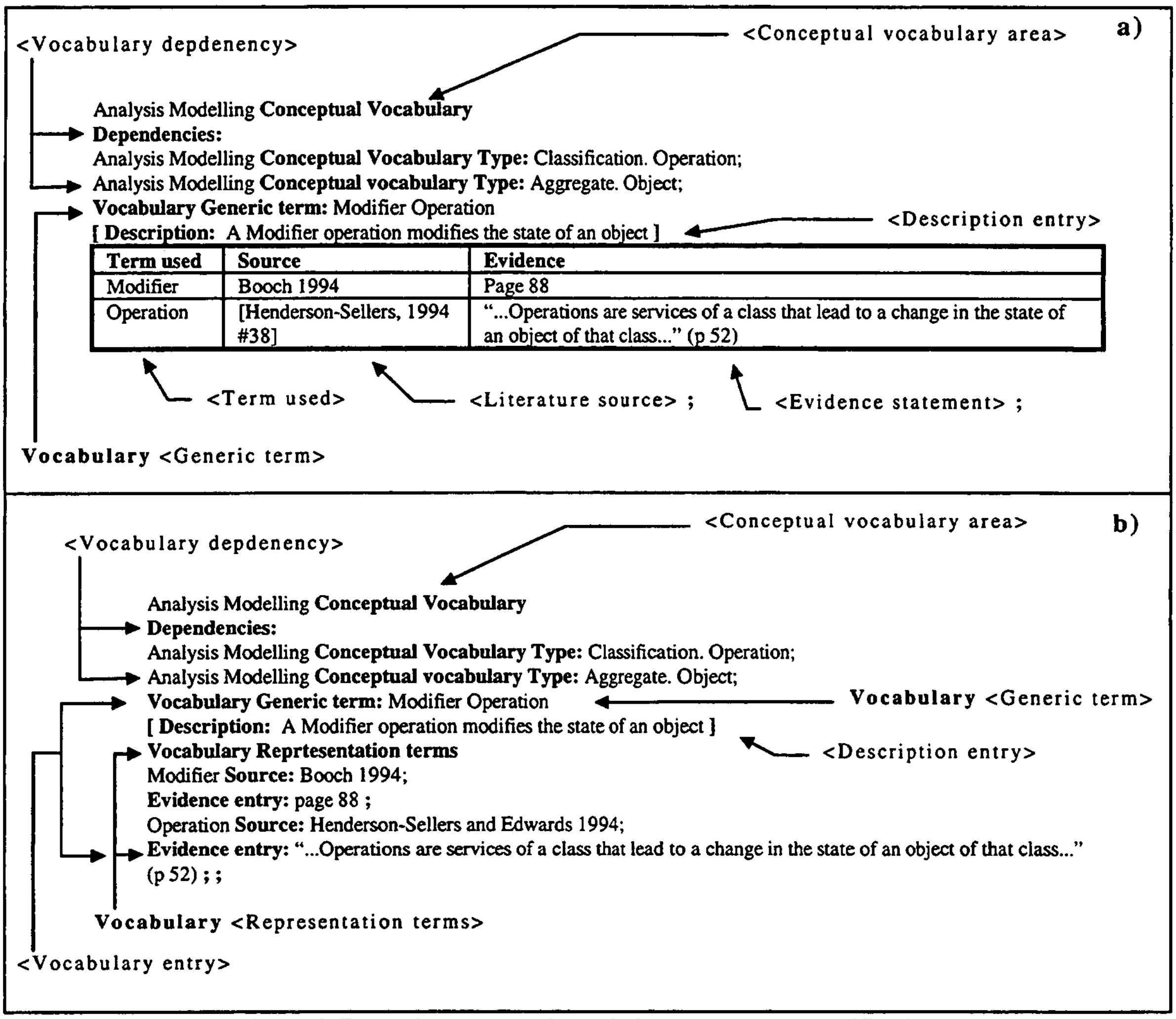


Figure 6: Example conceptual vocabulary component specification

Table 2 Features in Model Components (TDL Specifications)

Aggregate Features/Components	**Concept**	**Representation**	**Dependency**
Axis/ <Axis declaration>	<Generic term> <Description entry>	<Term used> <Evidence entry> "Source/Evidence"	N/A
Subdivision/ <Subdivision declaration>	<Generic term> <Description entry>	<Term used> <Evidence entry>	<Axis subdivision title> <Subdivision dependency>
Vocabulary/ <Conceptual vocabulary declaration>	<Vocabulary entry> <Generic term> <Description entry>	<Term used> <Evidence entry>	<Conceptual vocabulary area> <Vocabulary dependency>
Further Research/ <Further research declaration>	<Notes>	<Evidence entry>	<Further research title>
Literature/ <Literature source declaration>	N/A	<Evidence statement>	N/A

Table 3 Features in Model Components (Table Templates)

Aggregate Features/Components	**Concept**	**Representation**	**Dependency**
Axis/ *Axis Component*	<Axis name> <Description entry>	<Term used> <Literature source> <Evidence statement>	N/A
Subdivision/ *Subdivision Component*	<Generic term> <Description entry>	<Term used> <Literature source> <Evidence statement>	<Axis subdivision title> <Subdivision dependency>
Vocabulary/ *Conceptual Vocabulary component*	<Generic term> <Description entry>	<Term used> <Literature source> <Evidence statement>	<Conceptual vocabulary area> <Vocabulary dependency>
Further Research/ *Further research component*	<Notes>	<Literature source> <Evidence statement>	<Further research title>

The model component in Figure 6a) is usually described as a type (e.g the Conceptual Vocabulary Component), with examples described as instances (e.g. Analysis Modelling Conceptual Vocabulary). Other model component specifications use a similar pattern. ie One component has many instances similar to a structured process or object model having many application models; or the concept of a class in an object model (The type), with classes such as MyClass and QueueClass (The instances). In this work, model components are types, whereas instances are associated with one of these types. For example the axis component is the type, and the stages axis is an instance of the axis component or an axis instance. The illustrations in Figures 3 - 6 are all instances of model components.

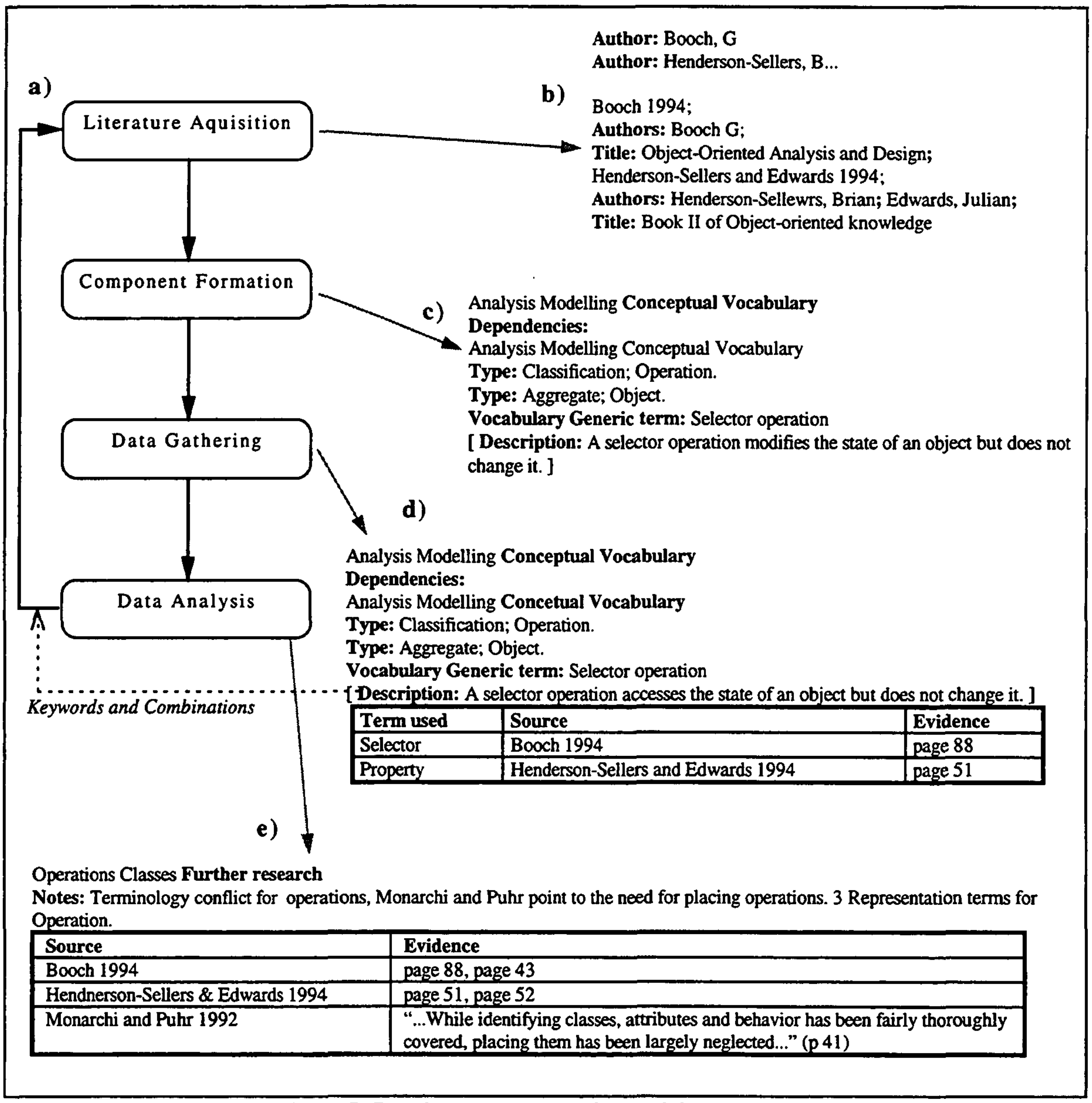
The TDL and table template specifications define the model components and their instances to some systematic degree. These specifications in Appendix A and Appendix B are like type definitions. The instances of these type definitions are like type variables (e.g. Figures 3 - 6). Each named component is a kind of model component, in the same way that a superclass has subclasses. For example, the axis and subdivision components are kinds of model components. The instances of model components together represent a research surveying tool in a subject for software methodologies (See Figure 1). The model components can be viewed as a meta-model of research surveying tools, with instances as particular research surveying tools. This is similar to meta-modelling in software methodologies (Doroshenko, 1997; Nissen, et al., 1996; Steele and Han, 1996), where an object model is a meta-model of application models and application models are instances of this meta-model.

DEVELOPMENT PROCESS AS A SOFTWARE LIFE CYCLE

The process for applying the model is illustrated in Figure 7a). The Literature Acquisition Stage is where relevant literature is identified and read based on a subject using standard methods (e.g. Leedy (1993) or Howard and Sharp (1983)). The purpose of this stage is to start the knowledge acquisition process in a subject and prepare the researcher for Component Formation. The products of this stage are instances of the literature source component. The literature source instances are similar to a bibliography of the literature. Two examples of this are shown in Figure 5b) and Figure 7b).

The Component Formation stage is where a number of dimensions, subdivisions, vocabulary areas, vocabulary concepts, and dependencies are identified as suitable for a research surveying tool. The purpose of this stage is to further knowledge acquisition of a subject of the researcher through the act of classification, and prepare a suitable framework for the Data Gathering stage. The products of this stage are draft axis, subdivision and conceptual vocabulary instances. The draft instances contain the names, dependencies, and interpretations, but no reference to identifiers for literature sources or representation terms (ie excluding instances of <Representation terms>, <Literature source>, <Term used>, and <Evidence statement>). An example of this is shown in Figure 7c) using the conceptual vocabulary component.

The Data Gathering stage is where literature sources are linked to axes, subdivisions, and conceptual vocabulary along with representation. The purpose of this stage is to further refine the researchers knowledge in a subject by classifying the literature using the instances and prepare the instances for the Data Analysis stage. By modelling the vocabulary and literature sources, researchers create a reference resource for revision via vocabulary and navigation based on literature and component instances. The products of this stage are complete axis, subdivision, and conceptual vocabulary instances. These are based on the draft instances from the Component Formation stage. Examples of this are shown in Figure 3, Figure 4, Figure 6, and Figure 7d).



Term used	Source	Evidence
Selector	Booch 1994	page 88
Property	Henderson-Sellers and Edwards 1994	page 51

Source	Evidence
Booch 1994	page 88, page 43
Hendnerson-Sellers & Edwards 1994	page 51, page 52
Monarchi and Puhr 1992	"...While identifying classes, attributes and behavior has been fairly thoroughly covered, placing them has been largely neglected..." (p 41)

Figure 7: Development process for model components

Maturity Indicators and Data Analysis

The Data Analysis stage is where component instances are analysed using a number of maturity indicators outlined in Table 4. The purpose of this stage is to identify where there may be candidate projects based on the maturity of the field or unresolved issues. The products of this stage are a number of numerical figures based on instances, along with instances of the further research component. Names, vocabulary areas, and vocabulary concepts support naming of project topics whereas maturity indicators along with further research instances help determine significance of project topics.

Maturity indicators are used to analyse instances of model components from the Data Gathering stage as a foundation for identification of candidate projects. In relating the level of maturity (Low, High) to the quantified indicator (Many, Few) it must be remembered that these are somewhat subjective terms which depend on the size of the research surveying tool. To get some objective idea of the relationship between field maturity and the indicators would require either an expert system or statistical analysis based on historical data. This is outside the scope (and space) of this paper. As a rule of thumb, few is less than ten, and many is ten or more.

Table 4: Maturity identification guidelines

Item	Identification in Model	Maturity Indication	
		Low	High
Literature Source Reference	<Literature Source> instance in Axis, Subdivision, & Conceptual vocabulary component instances	Few literature source references	Many literature source references
Conceptual Term	<Term> instance in <Generic term> instance in Axis, Subdivision, & Conceptual vocabulary component instances	Few conceptual terms	Many conceptual terms
Representation Term	<Term used> instance in Axis, Subdivision, & Conceptual vocabulary component instances	Many representation terms	Few representation terms
Dependency	<Subdivision dependency> instance or <Vocabulary dependency> instance	Few dependencies	Many dependencies
Homonym	Two component instances with differing <Generic term> and differing <Description entry> instances, have common <Term used> instance for a number of <Literature source> instances	Many homonyms	Few homonyms
Synonym	A Component instance with a given <Generic term> and associated <Description entry> instance, has a number of <Term used> instances for a number of <Literature source> instances	Many synonyms	Few synonyms
Concept Absence	A <Literature source> instance does not appear in an Axis, Subdivision, & Conceptual vocabulary component instance	Many incidences of concept absence	Few incidences of concept absence

Literature Source Reference: The Literature Source Reference maturity indicator is derived from instances of axis, subdivision and conceptual vocabulary components. By counting the number of literature source references, an indication of the work done based on the vocabulary is quantified. The value is obtained by counting the instances of <Literature source> in the model component's TDL specification, or counting the instances of <Literature source> in the "Source" column of the model component's table template specification. For example, in Figure 7d) there are two literature source references. Many literature source references indicate a high level of maturity, whereas few of these indicate a low level of maturity. For example, in Figure 7d) the number of Literature Source References would suggest that research related to the concept of selector operations for analysis modelling may be immature. Calculation of this maturity indicator may require removal of duplicates. The Literature Source Reference maturity indicator is a typical indicator of field maturity. For example, Monarchi and Puhr (1992) talk about the work done in placement and identification of operations. Wynekoop and Russo (1997) discuss work done based on research methods used (Action research, field experiment etc.) in studying software methodologies.

Conceptual Term: The Conceptual Term maturity indicator is derived from instances of axis, subdivision, and conceptual vocabulary components. By quantifying the size of the real vocabulary, as opposed to apparent terminology, an indication of the knowledge established in the field is quantified. The value is obtained by counting the instances of <Term> in instances of <Generic term> for the model component's TDL specification, or counting instances of <Term> in instances of <Generic term> in the model component's table template specification. For example, Figure 4a), b), and c) have three Conceptual Terms (Operation, Owns Aggregation, Identify Class). Many Conceptual Terms indicate a high level of maturity, whereas few of these indicate a low level of maturity. The Conceptual Term indicator points to the size of the real vocabulary in a subject.

Representation Term: The Representation Term maturity indicator is derived from instances of axis, subdivision, and conceptual vocabulary components. By quantifying the size of the apparent terminology, an indication of the level of awareness and consensus between researchers based on excessive vocabulary is indicated. The value is obtained by counting instances of <Term used> in the model component's TDL specification, or counting instances of <Term used> in the "Term used" column of the model component's table template specification. For example, based on Figure 4a), Figure 6a), and Figure 7d), there are seven Representation Terms. (Operation, Service, Property, Selector, Operation, Modifier, Behavior). Many Representation Terms indicate a low level of maturity, whereas few of these indicate a high level of maturity. The Representation Term indicator is one indicator that assess the volume of apparent vocabulary. Note that limiting application of this indicator to one axis name, subdivision name, or vocabulary concept yields the same result as the Synonym maturity indicator (See below). This indicator by itself is rarely useful and should be used in conjunction with other indicators.

Dependency: The Dependency maturity indicator is derived from instances of subdivision and conceptual vocabulary components. By counting the dependencies, an indication of the knowledge areas in a subject is quantified. The value is obtained by counting instances of "[Type: <Phrase>.] <Term>;" in the model component's TDL specification, or counting instances of "[Type: <Phrase>.] <Term>;" in the model component's table template specification. For example, based on Figure 4a), Figure 6a), and Figure 7d), there are five Dependencies. (operation and object, selector operation and object, modifier operation and object, selector operation and operation, modifier operation and operation). Many dependencies indicate a high level of maturity, whereas few of these indicate a low level of maturity. Dependencies reflect the amount of knowledge building that has occurred in a subject.

Homonym: The Homonym maturity indicator is derived from instance of axis, subdivision, and conceptual vocabulary components. By counting the number of homonyms, an indication of the level of consensus between researchers is quantified. The value is obtained by finding two instances of a <Generic term> in two model component's TDL specifications and comparing the instances of <Term used> that are equal in value, or by finding two instances of a <Generic term> in two model component's table specifications and comparing the instances of <Term used> in the "Term used" column that are equal in value. Each pair of values counts as one homonym. For example, looking at the <Term used> instances in Figure 6a) and Figure 4a), there is one Homonym (Operation). Many homonyms indicate a low level of maturity, whereas few of these indicate a high level of maturity. This indicator by itself is the most effective for assessing the level of maturity. Homonyms reflect the level of awareness between researchers and sometimes divergence and conflict in views about a subject.

Synonym: The Synonym maturity indicator is derived from instances of axis, subdivision and conceptual vocabulary components. By counting the number of synonyms, an indication of the level of awareness between researchers is quantified. The value is obtained by counting the instances of <Term used> for the same instance of <Generic term> in the model components TDL specification, or by counting the instances of <Term used> in the "Term used" column for the same <Generic term> in the model component's table template specification. For example, Figure 7d) has two Synonyms (Selector, Property). Synonyms reflect the level of awareness between researchers. In immature fields, researchers usually resort to introducing terminology that looks different, but may be the same as that of others.

Concept Absence: The Concept Absence maturity indicator is derived from instances of axis, subdivision, and conceptual vocabulary components. By counting concept absence, an indication of the division between research and divergence of views is quantified. The value is obtained by comparing the instances of <Literature source> in a model component's TDL specification to instances of <Literature source> in instances of the literature source component that are deemed relevant, or by comparing the instances of <Literature source> in the ""'Source" column of a model component's table template specification to instances of <Literature source> in instances of the literature source component that are deemed relevant. For example, Monarchi and Puhr (1992) are not included in Figure 7d) or Figure 6a), creating two Concept Absences. Many Concept Absences indicate a low level of maturity, whereas few of these indicate a high level of maturity. This indicator can converge with the Literature Source Reference indicator, but sources must be relevant. Determining relevance does rely on the judgement of the researcher. However, one rule of thumb is to use the instance of <Title> for the literature source instance as well as evidence from the text. For example, Booch (1994) and Shlaer and Mellor (1988) have "object-oriented" and "analysis" in their titles. Both sources also discuss modelling. Hence, they would be relevant to the Conceptual Vocabulary instance in Figure 6a) due to its vocabulary area ("Analysis Modelling"). Concept Absence reflects primarily the division of knowledge in a subject with more and more work becoming increasingly fragmented, particularly with journal and conference sources. Sometimes it can reflect a lack of work done in an area. However, this indicator by itself is not enough to assess maturity and other indicators should be used in conjunction with it.

Grouping and Calculation of Maturity Indicators

Although some indicators are better than others, no single maturity indicator can point to the maturity of the filed by itself. Hence, maturity indicators must be compared against each other to assess field maturity. For example, although a subject may have many instances of concept absence and representation terms. A high number of dependencies and conceptual terms, along with a low number of homonyms and synonyms really reflects an ordered knowledge building process. The consensus on views is indicated by consistent vocabulary (Homonym, synonym, conceptual term and indicators), and more and more people are building on previous work (Dependency indicator). A basic method is presented for obtaining and classifying values for maturity indicators suitable for analysis. Although it is not necessary to rely on statistical methods, they can be applied to the values. Numerous books on statistics can be found, for an introduction to the subject two sources are recommended (Huff, 1954; Moroney, 1965).

The maturity indicators are only applied to axis, subdivision, and conceptual vocabulary instances. Basic application of the maturity indicators is to sum the values for all instances. For example, using the instances in Figure 3, Figure 4, Figure 6a), and Figure 7d) the following values are obtained for the above maturity indicators: ten literature source references, eight conceptual terms, fourteen representation terms, nine dependencies, one homonym, nine concept absences, and for synonyms a mean of one point five, a median of two, and a mode of two. Figure 8a) illustrates presentation of data analysis for these results. Note that the synonym maturity indicator is one exception. It must be applied to each vocabulary concept, each axis name, and each subdivision name. To gain a single value for a number of instances, the mode, median, and mean calculations for the synonym maturity indicator are recommended. The values for the synonym maturity indicator are the values averaged, the number of instances are the number of samples (n).

Obtaining values for maturity indicators using all instances is not the only way to analyse the instances. Typically, values for indicators are partitioned based on the kinds of model components and their relationship to each other. One method for partitioning values is hierarchical selection. Figure 8 and Figure 9 show some results of this method. Values for indicators are partitioned based on the hierarchy of model components illustrated in Figure 2. To demonstrate this, the instances in Figure 3, Figure 4, Figure 6a), and Figure 7d) are used as if they represent an entire research surveying tool. The literature source reference maturity indicator is used for obtaining values.

a) Indicators for All Instances (The subject)

Indicator	Value
Literature Source References	10
Conceptual Terms	8
Representation Terms	14
Depedenedcies	9
Homonym	1
Concept Absences	9
Synonym	Mean:1.5, Median:2, Mode:2

b) All Axes

Indicator	Value
Literature Source Reference	4
(Other indicators)	Y

c) Axis Names

Axis Name→ Indicators↓	Stages	Components
Literature Source Refeneces	4	0
(Other indicators)	Y	Z

d) Stages Axis Name

Representation Term → Indicators↓	Phase	Stage	Activity
Literature Source Refeneces	2	1	1
(Other indicators)	X	Y	Z

Figure 8: Example data analysis for Subject and Axis Instances

A research surveying tool for a subject has a number of axes, subdivisions, and vocabulary areas (Figure 1). Values can be partitioned for all axes, all subdivisions, and all vocabulary areas. For example, there are four literature source references for all axes (Figure 8b)), two literature sources for all subdivisions, and five literature source references for all vocabulary areas (Figure 9a)). Where applicable, other indicators can be used in the same table (e.g. Figure 9a)).

An Axis has a name (Generic term), the name has a number of representation terms (Term used), a representation term is used in evidence from literature sources (Figure 1). Values can be obtained for each axis name and each representation term attached to an axis name. For example, using each axis name there are four literature sources for the stages axis, and zero literature source references for the components axis (Figure 8c)). Using each representation term attached to an axis name, the stages axis has two literature source references for the phases representation term, one for the stages representation term, and one for the activity term (Figure 8d)).
A Subdivision has a name (Generic term), the name has a number of representation terms (Term used), a representation term is used in evidence from literature sources (Figure 1). Values can also be obtained for each subdivision name, and each representation term attached to a subdivision name. For example, using all subdivisions for an axis, there are two literature source references for the phases axis. Using subdivision names there are two literature source references for the logical design subdivision. Using representation terms attached to a subdivision name, the logical design subdivision has two literature sources for the Logical design representation term.

a) All Vocabulary Areas

Indicator	Value
Literature Source Reference	5
Depdendency	8

b) Vocabulary Areas

Vocabulary Area→ Indicators↓	Analysis Modelling	Analysis Method Process
Literature Source Refeneces	5	1
Depdendency	6	2

c) Analysis Modelling Vocabulary Area

Vocabulary Name→ Indicators↓	Operation	Selector Operation
Literature Source Refeneces	3	2
Dependency	1	2

d) Analysis Modelling Vocabulary Area
Selector Operation Vocabulary Name

Representation Term→ Indicators↓	Selector	Property
Literature Source Refeneces	1	1
(Other indicators)	X	Y

Figure 9: Example Data Analysis for Vocabulary Area and Vocabulary Concepts

An Axis is composed of a number of subdivisions, a subdivision is used in a number of vocabulary areas, a vocabulary area has a number of vocabulary concepts (Generic term), a vocabulary concept has a number of representation terms (Term used), a representation term is used in evidence from literature sources (Figure 1). Values can be obtained for all subdivisions attached to an axis, each vocabulary area, each vocabulary concept attached to a vocabulary area, and each representation term attached to a vocabulary concept in a vocabulary area. For example, using each vocabulary area the Analysis Modelling vocabulary area has five literature source references, the method process vocabulary area has one literature source reference (Figure 9b)). Using each vocabulary concept attached to a vocabulary area, the operation vocabulary concept in the analysis modelling vocabulary area has three literature source references (Figure 9c)). Using representation terms attached to a vocabulary concept in a vocabulary area, the selector vocabulary concept in the analysis modelling vocabulary area has one literature source reference for the selector representation term, and one for the property representation term (Figure 9d)). Not all maturity indicators can be applied effectively at all levels for hierarchical selection. Appendix E Table 10 recommends which indicators can be applied at the given hierarchy levels.
Further research instances are used to capture any significant results a researcher wants to highlight. For example See the "Notes" entry in Figure 7e). Analysis of a subject using the model and processes can indicate a high level of maturity in some places, and a low level of maturity in others. Literature sources attached to component instances that indicate evidence of a low level of maturity are likely candidates for further examination. For example, (Monarchi and Puhr, 1992) is one such candidate (Figure 7e)). Further research instances can also

specify any information found about unresolved issues. For example, the (Monarchi and Puhr, 1992) entry in Figure 7e) discuss the lack of research in placement of classes, attributes and Behavior. Candidate issues identified by the researcher can also be included. The "Notes:" entry in Figure 5a) is one example.

Early in the research process, iteration from component formation to data analysis may result if a research surveying tool cannot provide adequate results. In these circumstances, use instances of representation terms (<Term used>) as keywords and instances of dependencies (Table 2, Dependency column) to guide keyword combinations for literature acquisition. For example, based on Figure 6a), Figure 7d), and Figure 7e), "identification AND (selector OR property) AND (operation OR service OR method)" could be a search string entered into a bibliography search engine such as Ovidweb Current Contents®.

To assist formation of topics for candidate projects, the vocabulary can be used in combination with further research issues previously identified. For example, Monarchi and Puhr (1992) raise the issue of research in placement of classes, attributes and behavior. This can be extended into projects for placement of owns aggregation from the instance in Figure 4b), or selector and modifier operations from the instances in Figure 6a) and Figure 7d). Research surveying, if time permits, can result in a large number of candidate projects offered to other researchers in addition to the researcher doing the surveying, further assisting the task of project identification. If an application of results can be found (e.g Fuggetta (1993)), or the size of the survey identifies a significant number of issues (e.g Doroshenko (1994)), the survey tool can become the project for an honours student. For PhD and Masters students, such a project can be part of a thesis based on some issues identified using the model and above processes. Vocabulary conflict can also be a foundation for a project (Examples can be found in four sources (Barki, et al., 1993; Doroshenko, 1994; Snyder, 1993; Soley and Stone, 1995)).

ANALOGICAL VIEW OF THE DEVELOPMENT PROCESS

The development process can be viewed as a software life cycle with modelling of a problem and evaluation of the proposed model to verify its usefulness for solving a problem. In this development process the model components are used to make a research surveying tool. Literature acquisition and component formation for a subject are similar to domain analysis of a problem with the resulting model. Data gathering and data analysis are analogous to an evaluation process to apply and evaluate the model. The surveying tool is evaluated to see if it generates useful results in the same way a software model is evaluated to see if it meets requirements. Iteration occurs if the model components do not generate useful results. This is similar to iteration occurring in software development if requirements are not fulfilled.

DISCUSSION

Dimension has much use in classifying software methodology aspects (See Table 1). Incorporation of these dimensions can reduce time in component formation, but some conflict in views may arise. e.g. Iivari (1995) has two dimensions for object-oriented analysis, compared to Lowry, et al. (1994), who have software development methodologies as one dimension. One possible solution is to describe dimension as a pattern in terms of subdivisions and dependency (The example is based on Henderson-Sellers and Edwards (1990)). A subdivision (phase) has classification dependency with a number of subdivisions below it (analysis, design). These subdivisions have aggregate dependency with subdivisions below themselves (user req specification, user req analysis; logical design, physical design). The axis is a projection, with the axis name being the top most subdivision (phase), with the classification dependency subdivisions attached to it (analysis, design etc). Although this simplifies the model and still allows inclusion of dimensions from other FMs, maturity indicators for data analysis are effected by such a change. Specifically, there would be no calculations for the axis, and the value for dependencies would increase and possibly change the interpretation of the data.

CONCLUSION

A model is presented to assist research surveying activities of researchers in software methodologies. Knowledge acquisition of a subject is assisted by classifying it using the model components, resulting in a research surveying tool. This tool can assist project identification by subjecting it to processes that indicate the maturity level in an area, examining more literature in that area, capturing research issues identified in the literature, and using the issues as a basis for candidate projects. These projects can contribute to department research if supervisors offer relevant subjects for research surveying. The model components and processes reduce a student's dependence on supervisor knowledge about research surveying activities.

Some issues are still open for inquiry. Firstly, the specification in Doroshenko (1994) provides much more data for classifying aspects of software methodologies. How these specifications relate to research surveying activities is not discussed in this paper. Secondly, although some papers in other IS research areas suggest evidence of model components (e.g. classification dependency and subdivision in Pervan (1994, pp 563 - 564), dimension and subdivision in Berci (1996, p 61)), successful application of the model in these areas for research surveying or in general is unknown. Thirdly, if the analogy is reasonably accurate, then automation of the model should be possible. This is similar to CASE technology support for developing software. Fourthly, there are two fundamental risks in using this approach to research surveying. Creation of a research surveying tool that is too large and contains data that is comprehensive but obsolete (An old phone book), or one that is tool small and contains data that is recent but trivial (A gossip magazine). This paper does not deal with management of these risks. Fifthly, although the model can support and corroborate a research topic, it cannot provide the "initial bright idea". Instances of model components must still be formed by the researcher in a given subject (The author doesn't know of a software model that can build itself). This requires imagination and creativity, both of which are a part of research work (Research, 1997; Whitehead, 1929).

REFERENCES

Backus, J. W. (1959) "The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference", , **International Conference on Information Processing**, Paris, France, Unesco, pp 125 - 132.

Barki, H., Rivard, S., andTalbot, J. (1993) "A Keyword Classification Scheme for IS Research Literature", **MIS Quarterly**, Vol , No , pp 209 - 226.

Berci, B. J. (1996) "Strategic Information Management vs. Strategic Business Management", , **Australasian Conference on Information Systems**, Hobart, Australia, Australian Computer Society Inc, pp 57 - 68.

Beringer, D. (1994) "Limits of Seamlessness in Object-oriented Software Development", , **TOOLS Europe**, , Prentice-Hall, pp 1 - 11.

Booch, G. (1994) **Object-Oriented Analysis and Design with Applications**, 2nd, Redwood City, Benjamin/Cummings,

Curtis, B., Kellner, M. I., andOver, J. (1992) "Process Modelling", **Communications of the ACM**, Vol 35, No 9, pp 75 - 90.

Dart, S. A., Ellison, R. J., Fieler, P. H., andHabermann, A. N. (1987) "Software Development Environments", **IEEE Computer**, Vol , No 11, pp 18 - 28.

De Champeaux, D. and Faure, P. (1992) "A comparitive study of object-oriented analysis methods", **Journal of Object-Oriented Programming**, Vol , No March/April, pp 21 - 32.

Doroshenko, E. E. (1994) **Common Concepts in Object-oriented Methodologies**, Applied Computing & Mathematics, University of Tasmania.

Doroshenko, E. E. (1996) "Toward a language-graphic model for CASE tool construction: helping the corner store OT vendor", , **Australasian Conference on Information Systems**, Hobart, Australia, Australian Computer Society Inc, pp 161 - 172.

Doroshenko, E. E. (1997) "A Refined Meta-Model Approach to Education for the Transition to Object Technology", , **International Conference on Object-Oriented Information Systems**, Brisbane Australia, Springer-Verlag, pp 466 - 476.

Fernstrom, C., Narfelt, K.-H., andOhlsson, L. (1992) "Software Factory Principles, Architecture, and Experiments", **IEEE Software**, Vol 9, No 2, pp 36 - 44.

Fuggetta, A. (1993) "A Classification of CASE Technology", **IEEE Computer**, Vol 26, No 12, pp 25 - 38.

Henderson-Sellers, B. and Edwards, J. M. (1990) "The Object-Oriented Systems Life Cycle", **Communications of the ACM**, Vol 33, No 9, pp 142-159.

Howard, K. and Sharp, J. A. (1983) **The Management of a Student Research Project**, , Hampshire, Gower,

Huff, D. (1954) **How To Lie with Statistics**, , London, Victor Gollancz Ltd,

Iivari, J. (1995) "Object-orientation as structural, functional and behavioural modelling: a comparison of six methods for object-oriented analysis", **Information and Software Technology**, Vol 37, No 3, pp 155-163.

Korson, T. D. and Vaishnavi, V. K. (1992) "Managing Emerging Software Technologies: A Technology Transfer Framework", **Communications of the ACM**, Vol 35, No 9, pp 101 - 111.

Leedy, P. D. (1993) **Practicl Research: Planning and Design**, 5th, New York, MacMillan,

Lowry, G. R., Godfrey, R., Lear, F. A., andMorgan, G. W. (1994) "A 4th Generation Information Systems Development Research Framework", , **5th Australasian Conference on Information Systems**, Melbourne Australia, Department of Information Systems, Monash University, Australia, pp 221-236.

McChesney, I. R. (1995) "Toward a classification scheme for software process modelling approaches", **Information and Software Technology**, Vol 37, No 7, pp 363 - 374.

Monarchi, D. E. and Puhr, G. I. (1992) "A Research Typology for Object-Oriented Analysis and Design", **Communications of the ACM**, Vol 35, No 9, pp 35-47.

Moroney, M. J. (1965) **Facts from Figures**, 2nd, London, Penguin Books,

Naur, P. (1963) "Revised report on the algorithmic language Algol", **Communications of the ACM**, Vol 6, No 1, pp 1-17.

Nissen, H. W., Jeusfeld, M. A., Jarke, M., Zemanek, G. V., andHuber, H. (1996) "Managing Multiple Requirements Perspectives with Metamodels", **IEEE Software**, Vol 13, No 2, pp 37 - 47.

OMG, O. M. G. (1996) **Object Analysis & Design Facility RFP-1**, Farmingham MA, Object Management Group.

Pervan, G. P. (1994) "The Measurement of GSS Effectiveness: A Meta-analysis of the Literature and Recommendations for Future GSS Research", , **Hawaii International Conference on Systems Sciences**, Maui, Hawaii, , pp 562 - 571.

Research, O. f. (1997) **Research Higher Degrees Handbook**, , Hobart, University of Tasmania,

Roper, N. (1995) **Common Concepts in Structured Process Methodologies**, Applied Computing & Mathematics, University of Tasmania.

Sethi, R. (1989) **Programming Languages: Concepts and Constructs**, , Reading, MA, Addison-Wesley,

Shlaer, S. and Mellor, S. J. (1988) **Object-Oriented Systems Analysis: Modelling the World in Data.**, , Englewood Cliffs, New Jersey, Prentice-Hall,

Snyder, A. (1993) "The Essence of Objects: Concepts and Terms", **IEEE Software**, Vol 10, No 1, pp 31 - 42.

Soley, R. M. E. and Stone, C. M. (1995) **Object Management Architecture Guide**, 3, New York, Wiley & Sons,

Steele, P. M. and Han, J. (1996) "A Layered Architecture for Describing Information Systems Development Methodologies", , **Australasian Conference on Information Systems**, Hobart, Australia, Australian Computer Society, pp 677-688.

Whitehead, A. N. (1929) **The Aims of Education**, , New York, MacMillan Company,

Wynekoop, J. L. and Russo, N. N. (1997) "Studying system development methodologies: an examination of research methods", **Information Systems Journal**, Vol 7, No 1, pp 47 - 65.

ACKNOWLEDGMENTS

Eugene Doroshenko would like to thank his honours supervisor, Ms Felicity Lear, PhD supervisor, Professor Glenn Lowry, the Scolarships Officer, Vanessa Folvig, the Head of School, Professor Chris Keen, and Professor Robert Delbourgo, for their assistance at the University of Tasmania.

APPENDICES

Appendix A

Component Specifications

<Axis declaration> ::= { <Axis name> <Description entry> <Axis representation> }
<Axis name> ::= **Axis** <Generic term>
<Axis representation> ::= **Axis** <Representation terms>
- <Term> **in** <Axis name> **unique within** <Axis declaration>

<Subdivision declaration> ::= { <Axis subdivision title> <Subdivision dependency> **Subdivisions:** { <Subdivision entry> } }
<Axis subdivision title> ::= <Term> **Axis.**
- <Term> **in** <Axis subdivision title> **there must exist** <Term> **in** <Axis name> **in** <Axis declaration>

<Subdivision dependency> ::= [**Dependencies:** { [**Type:** <Phrase> **.**] <Term> **;** }]
- <Term> **in** <Subdivision dependency> **there must exist** <Term> **in** <Subdivision entry> **in** <Subdivision declaration>

<Subdivision entry> ::= **Subdivision** <Generic term> <Description entry> **Subdivision** <Representation terms>
- <Term> **in** <Generic term> **in** <Subdivision entry> ≠ <Term> **in** <Axis title> **in** <Axis declaration>
- <Term> **in** <Generic term> **in** <Subdivision entry> **unique within** <Subdivision declaration>

<Conceptual vocabulary declaration> ::= { <Conceptual vocabulary area> <Vocabulary dependency> { <Vocabulary entry> } }
<Conceptual vocabulary area> ::= <Title> **Conceptual Vocabulary.**
<Vocabulary dependency> ::= [**Dependencies:** { <Conceptual vocabulary area> { [**Type:** <Phrase>] <Term> **;** } }]
- (<Conceptual vocabulary area> **AND** <Term>) **in** <Vocabulary dependency> **there must exist** (<Term> **in** <Generic term> **AND** <Conceptual vocabulary area>) **in** <Conceptual vocabulary declaration>
- <Term> **in** <Generic term> **in** <Vocabulary entry> ≠ <Term> **in** <Conceptual vocabulary dependency>

<Vocabulary entry> ::= **Vocabulary** <Generic term> <Description entry> **Vocabulary** <Representation terms> **;**
- <Conceptual vocabulary area> (<Term> **in** <Generic term> **in** <Vocabulary entry>) **unique**

<Further research declaration> ::= { <Further research area> <Further research entry> <Notes> }
<Further research entry> ::= **Issue evidence:** { <Evidence entry> }
<Further research area> ::= <Term> { <Term> } **Further research**
<Notes> ::= **Notes:** [<Phrase>]
- <Term> **in** <Further research title> **there must exist** (<Term> **in** <Generic term> in <Subdivision entry>) OR (<Term> **in** <Generic term> in <Axis title>) **OR** (<Term> **in** <Generic term> **in** <Vocabulary entry>)

<Literature source declaration> ::= { <Author declaration> } { <Literature source> ; <Literature source entry>**.** }
<Literature source> ::= <Author last name> [((**and** | **&**) <Author last name>) | (**et al**))] <Year> { <Letter> }
<Literature source entry> ::= **Authors:** { <Author entry> **;** } <Source title> <Source type> <Year published> <Publisher> **Evidence:** { <Evidence statement> } **;**
<Source type> ::= **Type:** (**Book;** <Book details>) | (**Journal article;** <Journal article details>) | (**Conference paper;** <Conference paper details>) | (**Other;** <Informal>)
<Author declaration> ::= { <Author entry> **;** }
<Author entry> ::= **Author:** <Author last name> **,** <Author first name> [<Author middle name>]
<Year published> ::= **Year:** <Year> **;**
<Source title> ::= **Title:** <Phrase> **;**
<Publisher> ::= **Publisher:** <Phrase> **;**
<Year> ::= <Number> [**AD** | **BC** | **BCE**] **;**
- <Literature source> **in** <Literature source declaration> **unique within** <Literature source declaration>
- <Author entry> **in** <Literature source entry> **there must exist** <Author entry> **in** <Author declaration>
- <Author last name> **in** <Literature source> **there must exist** <Author last name> **in** <Author entry> **in** <Literature source entry>

- <Year> **in** <Literature source> **there must exist** <Year> **in** <Year published> **in** <Literature source entry>

Universal Terminals and Non-terminals

<Description entry> ::= '[' **Description:** { <Informal> | <Evidence entry> } ']'
<Title> ::= <Term> { <Term> }
<Generic term> ::= **Generic term:** <Term>
<Representation terms> ::= **Representation terms:** [<Term used> { <Evidence entry> } { , <Term used> { <Evidence entry> } }] ;
<Evidence entry> ::= **Source:** <Literature source> **Evidence entry:** { <Evidence statement> } ;
<Evidence statement> ::= (<Figure reference> | <Quotation> | <Page reference> | <Phrase> | <Informal>) [. | ,]

- <Term> **in** <Title> **there must exist** <Term> **in** <Subdivision entry>
- (<Literature source> **AND** <Evidence statement>) **in** <Evidence entry> **there must exist** (<Literature source> **AND** <Evidence statement> **in** <Literature source entry>) **in** <Literature source declaration>

<Term used> ::= <Phrase> | ?

Appendix B

NB The Table Templates make use of some terminals and non-terminals found in Appendix A. The Literature Source Component does not have a Table Template specification

Axis Component

<Axis name>
<Description entry>

Term used	**Source**	**Evidence**
<Term used>	{ <Literature source> ; }	{ <Evidence statement> ; }

Subdivision Component

<Axis subdivision title>
<Subdivision dependency>
Subdivision <Generic term>
<Description entry>

Term used	**Source**	**Evidence**
<Term used>	{ <Literature source> ; }	{ <Evidence statement> ; }

Conceptual Vocabulary Component

<Conceptual vocabulary area>
<Vocabulary dependency>
Vocabulary <Generic term>
<Description entry>

Term used	**Source**	**Evidence**
<Term used>	{ <Literature source> ; }	{ <Evidence statement> ; }

Further Research Component

<Further research area>
<Notes>

Source	**Evidence**
<Literature source>	<Evidence statement>

Appendix C

Table 5: Subdivision Component and Specification

Component Item	TDL	Table Template
Subdivision Component	<Subdivision declaration>	Subdivision Component
Axis Link to Subdivisions	<Axis subdivision title>	<Axis subdivision title>
Name	<Generic term> in <Subdivision entry>	<Generic term>
Interpretation	<Description entry>	<Description entry>
Dependency	<Subdivision dependency>	<Subdivision dependency>
Representation Terms	<Subdivision entry> (<Term used> in <Representation terms>)	<Term used>
Literature Source for Representation Term	<Literature source> in <Evidence entry>	<Literature source>
Evidence in Literature Source	<Evidence statement> in <Evidence entry>	<Evidence statement>

Table 6: Axis Component and Specification

Component Item	TDL	Table Template
Axis Component	<Axis declaration>	Axis Component
Name	<Axis name>	<Axis name>
Interpretation	<Description entry>	<Description entry>
Representation Terms	<Axis representation> (<Term used> in <Representation terms>)	<Term used>
Literature Source for Representation Term	<Literature source> in <Evidence entry>	<Literature source>
Evidence in Literature Source	<Evidence statement> in <Evidence entry>	<Evidence statement>

Table 7: Conceptual Vocabulary Component and Specification

Component Item	TDL	Table Template
Conceptual Vocabulary	<Conceptual vocabulary declaration>	Conceptual Vocabulary Component
Vocabulary Area	<Conceptual vocabulary area>	<Conceptual vocabulary area>
Vocabulary Concepts	<Vocabulary entry>	One instance per concept
Name for Vocabulary Concept	<Generic term> in <Vocabulary entry>	<Generic term>
Interpretation for Vocabulary Concept	<Description entry> in <Vocabulary entry>	<Description entry>
Representation Terms	<Vocabulary entry> (<Term used> in <Representation terms>)	<Term used>
Literature Source for Representation Term	<Literature source> in <Evidence entry>	<Literature source>
Evidence in Literature Source	<Evidence statement> in <Evidence entry>	<Evidence statement>
Dependency	<Vocabulary dependency>	<Vocabulary dependency>

Table 8: Further Research Component and Specification

Component Item	TDL	Table Template
Further research component	<Further research declaration>	Further Research Component
Further research component Link to Axis, Subdivision, Conceptual Vocabulary Area, Vocabulary Concept(s)	<Further research area>	<Further research area>
Issue	<Notes>	<Notes>
Literature source	<Literature source> in <Evidence entry>	<Literature source>
Evidence from Literature Source	<Evidence statement> in <Evidence entry>	<Evidence statement>

Table 9: Literature Source Component and Specification

Component Item	TDL	Table Template
Literature source component	<Literature source declaration>	N/A
Author	<Author declaration>	N/A
Literature source	<Literature source entry>	N/A
Identifier	<Literature source>	N/A
Literature source Link to Author(s)	<Author entry> in <Literature source entry>	N/A
Type	<Source type> in <Literature source entry>	N/A
Literature Source Link to Evidence Extracts	<Evidence statement> in <Literature source entry>	N/A

Appendix D

The TDL is used to describe itself. Non-terminals are enclosed by angle brackets (<A non-terminal>), terminals are written in bold face (**A_terminal**), production rules have a non-terminal name followed by "::=" and a series of non-terminals and terminals (<A non-terminal> ::= <A non-terminal> **A_terminal**). Braces represent zero or more options for terminals and non-terminals ({ **A_Terminal** <A Non-terminal> }), brackets represent an optional item ([<A non-terminal>]) parentheses are used to group items ((<A non-terminal> **A_terminal**)), a vertical bar represents a choice between the items (<A non-terminal> | **A_terminal**). Reserved symbols used as terminals are enclosed by single quotes ('[' or ')'). Relationships between instances of non-terminals can exist, called semantic statements (See table below). Each semantic statement is preceded by a "•". An element is a valid instance of a terminal/non-terminal sequence (e.g. "an_identifier" is an element of <Identifier>, where <Identifier> ::= <Letter> { <Letter> | <Digit> }).

Semantic statement	Purpose	Example Application
Existence semantic (<Existence semantic>)	Dependency between two elements	Type identifier in a variable declaration
Uniqueness semantic (<Uniqueness semantic>)	Uniqueness of elements	Global Variable identifier in a program
Alias semantic (<Alias semantic>)	Duplicate naming of same element	variable passed as actual parameter
Cardinality semantic (<Cardinality semantic>)	number of instances between two elements	a type identifier & variable identifiers
Equality semantic (value) (<Equality semantic>)	Value equality between two elements	type identifier and variable identifier
Equality semantic (number) (<Equality semantic>)	Number equality between two elements	number of type and variable identifiers
Enclosure semantic (<Enclosure semantic>)	Limits application of uniqueness	local variable in procedure declaration
Premise clause (<Premise clause>)	Contingent application of semantic statement	private keyword for class member

<Text definition statement> ::= { <Production rule> | <Semantic statement> } }
<Production rule> ::= <Non-terminal name> '::=' <Non-terminal structure>
<Non-terminal structure> ::= { '{' <Production rule item> '}' | '[' <Production rule item> ']' | '(' <Production rule item> ')' | (<Production rule item> ['|' <Production rule item>]) }
<Production rule item> ::= <Non-terminal structure> | <Non-terminal name> | <Terminal name> | <Notation definition statement> | <Diagram description>
<Non-terminal name> ::= '<' <Plain character> { <Plain character> } '>'
<Terminal name> ::= { ('" <Reserved symbol> "') | <Bold character> }
< Semantic statement> ::= • <Semantic type> { <Logic operator> <Semantic type> } .
<Semantic type> ::= [<Premise clause>] (<Existence semantic> | <Uniqueness semantic> | <Alias semantic> | <Equality semantic> | <Enclosure semantic>)
<Uniqueness semantic> ::= <Semantic reference> **unique** [**within** <Semantic reference>]
<Existence semantic> ::= <Semantic identification> **there must exist** <Semantic identification>
<Alias semantic> ::= <Semantic reference> **Alias for** <Semantic reference>
<Cardinality semantic> ::= **For** <Cardinality> { <Semantic reference> } **there are** <Cardinality> { <Semantic reference> }
<Equality semantic> ::= ['(' <Bound> ')'] <Semantic identification> <Operator> ['(' <Bound> ')'] <Semantic identification>
<Enclosure semantic> ::= <Semantic identification> **encloses** <Semantic identification>
<Operator> ::= '=' | '<' | '>' | ≤ | ≥ | ≠
<Premise clause> ::= **If** <Semantic identification> | <Semantic type> =>
<Semantic identification> ::= <Semantic reference> { <Logic operator> <Semantic reference> }
<Semantic reference> ::= (<Semantic element> { **in** <Semantic element> }) | ('(' <Semantic identification> ')')
<Semantic element> ::= { <Terminal name> | <Non-terminal name> | <Phrase> | <Instance identifier> } ['(' <Number> ')']
<Instance identifier>::= " { <Bold character> | <Space> } "
<Reserved symbol> ::= '{' | '}' | '<' | '>' | '|' | '(' | ')' | '::=' | '[' | ']' | '"' | '=>' | '•'
<Cardinality> ::= '(' <Bound> : <Bound> ')'
<Bound> ::= <Capital Letter> | <Number>

- <Non-terminal name> **in** <Production rule> **unique within** <Text specification definition statement>

Appendix E

Table 10: Partitioning of values for Hierarchical Selection

Indicators → *Level ↓*	Literature Source reference	Conceptua l Term	Rep. Term	Depdend.	Homonym	Synonym	Concept Absence
Axis	X	X	X		X	X	X
Axis name	X	X	X: Same as Synonym			X	X
Axis Rep. term	X						
Subdivision	X	X		X	X	X	X
Subdivision name	X	X	X: Same as Synonym	X		X	X
Subdivision rep. term	X						
Vocabulary Areas	X	X		X	X	X	
Vocabulary Area name	X	X		X	X	X	
Vocabulary Concept	X	X	X: Same as Synonym	X		X	X
Vocabulary rep. term	X						