

DIVISION OF COMPUTER SCIENCE

**An Investigation into Measurement of Notations Used in
Software Modelling**

**Carol Britton
Sara Jones
Margaret Myers
Mitra Sharif**

Technical Report No. 257

September 1996

An investigation into measurement of notations used in software modelling

Carol Britton¹, Sara Jones¹, Margaret Myers², Mitra Sharif¹

¹ School of Information Sciences,
University of Hertfordshire, Hatfield, AL10 9A, UK
Tel: +44-(0)1707-284321, Fax: +44-(0)1707-284303
EMail: C.Britton, S.Jones, M.Shariff@herts.ac.uk

² Department of Computing, Science and Mathematics,
Richmond, The American International University in London,
Queens Road, Richmond, Surrey, TW10 6JP
Tel: +44-(0)181-455-7921, Email: myersm@staff.richmond.ac.uk

1 Introduction

Measurement systems for software abound in the literature as do those for models in specification languages. This report applies measurement to the formal notations themselves making use of the principles of model-based measurement. Formality implies the use of mathematics. Any notation which relies on a branch of mathematics is therefore formal to a lesser or greater degree depending on the scope of the branch of mathematics chosen¹ and whether a deductive mechanism is included to allow reasoning on the model.

WIE95 points out the difficulties of integrating various modelling schemes and even of knowing whether they can be integrated. A well established engineering principle is that without measurement there is no knowledge. We therefore set out to investigate whether there is a way of establishing a measurement scheme for modelling notations. The development of such a measurement scheme would be helpful in two ways: setting up the scheme leads to a deeper understanding of individual formal methods while the measurement values themselves would allow comparisons to be made between the various notations. Such comparisons could provide objective information for choice of individual methods and suggest suitable combinations of such methods for specific purposes.

The rest of Section 1 provides guidance in the approach adopted: model-based measurement, acquisition of measures, quality of measure. Section 2 summarizes elements of the systems approach introduced in KAP94 and now included in system Studies at Richmond.

Sections 3, 4 and 5 are devoted to direct measures. Section 3 applies the systems approach to notations modelled as attribute/relation pairs. The emphasis is only on representing attributes for which measures can be suggested. Section 4 shows how structural models can show relations between the tokens, relations such as `can_precede`, `can_follow`. Section 5 introduces the notion of scalability, deriving attributes of the whole from the parts and the relations between the parts and also its converse: increasing granularity to allow more sensitive measures of the atomic components of a structure.

Section 6 suggests possible indirect measures. The section is short and conjectural.

Section 7 summarizes the work undertaken for this Report and suggests further development.

¹ PRE94 refers to 'a spectrum of formality'.

1.1 Model-based measurement

Models are *finite, purposeful, abstractions* of real world objects represented in a formal notation. Model-based measurement is a two-stage process (figure 1.1). The first stage creates the model, the second obtains values for particularising the model.

Modelling involves the identification of attributes of interest of the real world object, the referent. In this case the referent is the formal notation itself. Each of the chosen attributes is represented as a property variable and the set of variables is the model of the referent class: this group of attribute values is *finite*, an abstraction of the referent's possible infinite number of attributes, intentionally or otherwise, the chosen group is an abstraction of the totality of attributes, *purpose* is assumed (else why carry out the process) and the notation is that of mathematics, definitely formal.

Measurement, the second stage of the process, consists of allocating values to the property variables. Now the model represents only the individual referent on which the measurement is carried out. Note the implicit assumption in this representation, that all the chosen attributes belong to the same referent. In this model, no account is taken of the internal structure of the referent. This type of model is known as a *black-box* model.

A simple example illustrates the above. An estate agent may model the class of houses in terms of attributes she thinks are of interest to the buyer (purpose). Four attributes (finite) are initially chosen (abstraction) and modelled as properties:

Number of bedrooms N_b
Number of reception rooms N_r
Address A
Price P

Each house on the agent's books is modelled by a set of 4 property-name/property-value pairs composed as above. No two models are alike since an address is unique.....**to be continued**

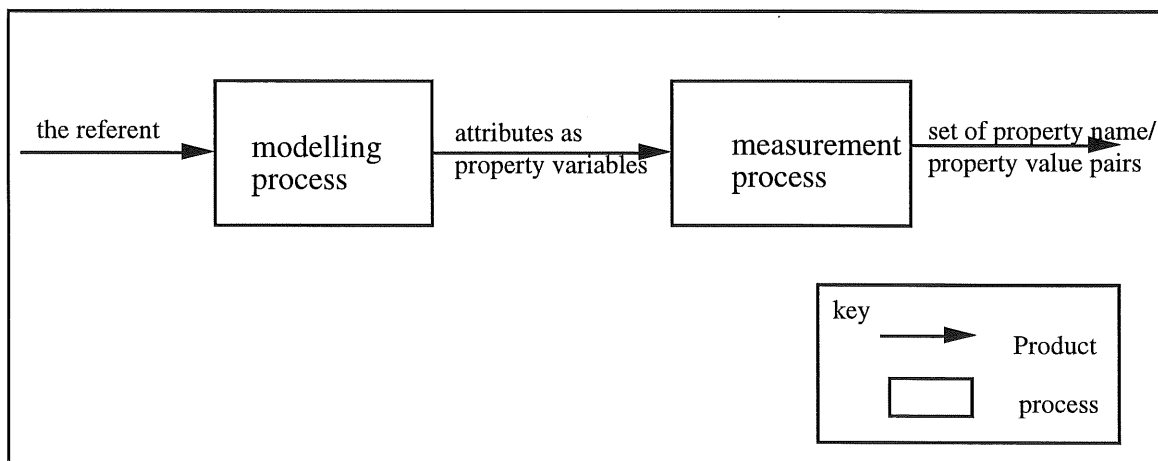


Figure 1.1: Model based measurement

1.2 Obtaining measurement values

Direct and indirect measures: Values of measures may be obtained directly by inspection, comparison with a standard or instrumentation. Values not directly available may be deduced using domain theory: the relation between depth, breadth and area, volts, amps and watts, etc. Whether a value is obtained directly or indirectly depends only on the nature of the measure and the instrumentation available.

Object centred measures: A set of measures, all belonging to the same referent are 'object centred'. They characterise the referent and these property-name/property value pairs form the entity set of the *system model* to be introduced in Section 2.2.

Utility measures: Where value judgements are to be made on a referent, then utility measures are defined in which the judge's subjective values are made explicit: the level of noise to be allowed at an airport may be judged to be dependent on the population density, for instance. Utility measures stand at the pinnacle of the hierarchy of figure 1.2. Below that level all measures are objective and verifiable to the required or stated degree of accuracy. Only at the final stage may the observer's views be incorporated. Note that no utility attribute measure is acceptable if it is not defined in terms of measurable attributes.

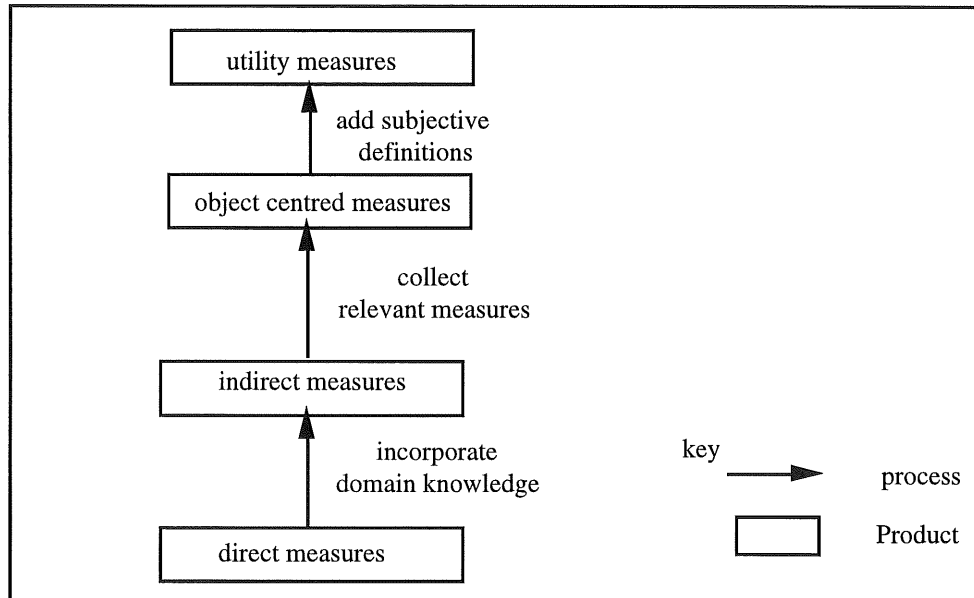


Figure 1.2: Measurement hierarchy

1.3 General obligations of the measurer

The measurer must first describe each attribute either in terms agreed by consensus among the relevant scientific community or define it in terms of such concepts. For direct and indirect measures, the measurer must show how the attribute is to be measured objectively, on what scale and in what dimensional units and to what order of accuracy. Finally, for utility measures, the measurer must state and justify assumptions upon which the measure is based [KAP94]. The results of calculating and applying utility measures to real situations confirms or invalidates these assumptions.

2 Referents and their System Models

2.1 The referents as systems

We use systems notation throughout. This states that any purposeful structure of parts forms a *system*. A system is thus a pair, a set of entities (E), and a set of relations (R) over those entities. A system is thus:

$$\begin{aligned}
 S &= (E, R) \\
 \text{where } E &= \{e_1, e_2 \dots\} \text{ and } R = \{r_1, r_2 \dots\} \\
 &\quad e_1, e_2 \dots \text{ are entities} \\
 \text{and } &\quad r_1, r_2 \dots \text{ are relations.}
 \end{aligned}
 \tag{1}$$

expression (1)

Any written or verbal description of a referent, a real-world entity, is a representation. Such a representation, which may or may not be a model, is finite. We may therefore replace the infinite sets of expression (1) by the bounded sets shown below.

$$S = (E, R) \\ \text{where } E = \{e_1, e_2 \dots e_n\} \text{ and } R = \{r_1, r_2 \dots r_n\} \quad \text{expression (2)}$$

and e and r are as before.

2.2 Black Box models of referents

The black box system representation of a referent has attributes as its entities. The minimum relations set consists of a single co-attribute relation which confirms that all the enumerated attributes belong to the same referent.

The black box system model of a referent replaces each attribute by a property name/property value pair and adds a time stamp relation to the relation set which confirms the time at which the measured values are obtained. Where the property value is a variable, then the model represents the whole **class** of referents which share the properties. To summarise: the entity set of the black box model is the set of property-name/property-value pairs comprising the object centred measures of the referent. The relation set contains at least two relations: the co-attribute relation affirming that all the measures belong to the same referent and the co-temporal relation or time stamp representing the time at which the measures were obtained.

We continue with the example from Section 1.2. Collecting the 4 property name/property variables into the entity set and adding the two mandatory relations, the systems model of a house for the purpose set by the Estate Agent is:

$$S_{\text{(house)}} = (\{\text{number_of_bedrooms}/N_b, \text{Number_of_receptn_rooms}/N_r, \text{Address}/A, \text{Price}/P\} \\ \{r_0, r_t\}).$$

Note the presence of the time stamp r_t . Attributes such as price may change, even number of rooms if the Council allows. No set of attributes preserves its integrity over a long time period and this is plain in the model.

2.3 Structural models of referents

The Structural system representation of any referent has, as its entities, the referent's component parts, each part represented as a black box. The relation set contains the structural relations describing how the components are joined while a co-attribute relation affirms that all the components belong to the same structural system model. All the component black box models within the entity set must contain the same time stamp which also represents the time stamp of the system model as a whole (figure 2.1).

The double headed arrow in figure 2.1 and labelled behavioural equivalence expresses the fact that both the black box model symbolised by the root black box and the structural subtree emanating from it have the *same* referent and the *same* choice of attributes being modelled. An initial choice of 'same referent' in the key was therefore replaced by the current label.

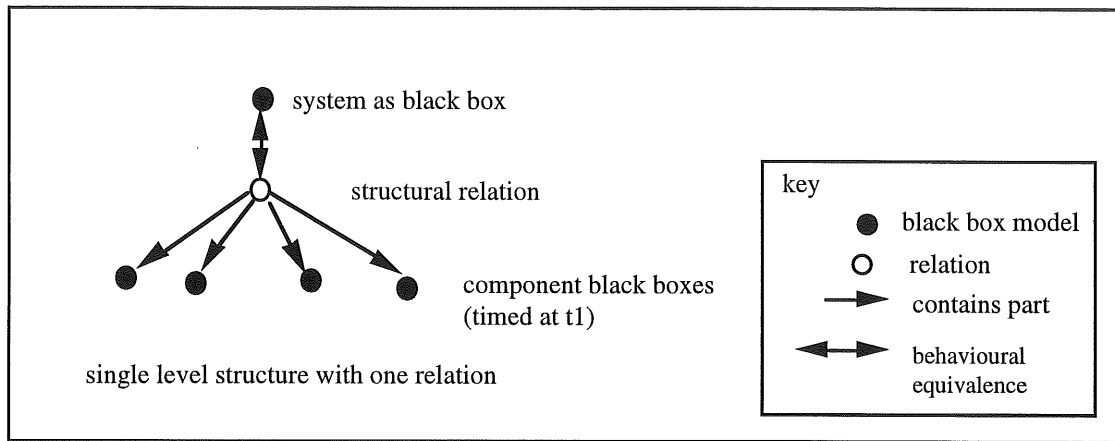


Figure 2.1: Structural system models

The top level node of figure 2.1 is the black box system model of the structure, hence its property values should be derivable from the property values of each of its components and the relations combining those components. Figure 2.2 shows a multilevel structure where one of the black box models of figure 2.1 has been refined further.

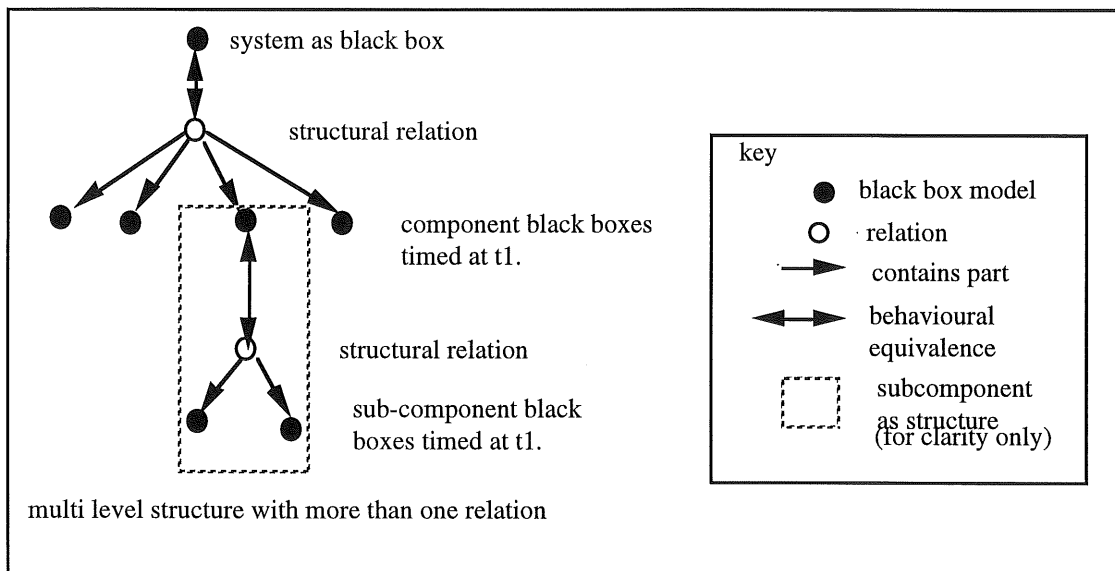


Figure 2.2: Structural system models

Figure 2.3 shows the derivation for a single property, resistance, where the structural relation is 'in_parallel' for two resistors, r_1 and r_2 . In general, each single property of the whole should be derivable from all the properties of the components and the domain theory, where known. The relation over the attributes will depend on the nature of that particular attribute.

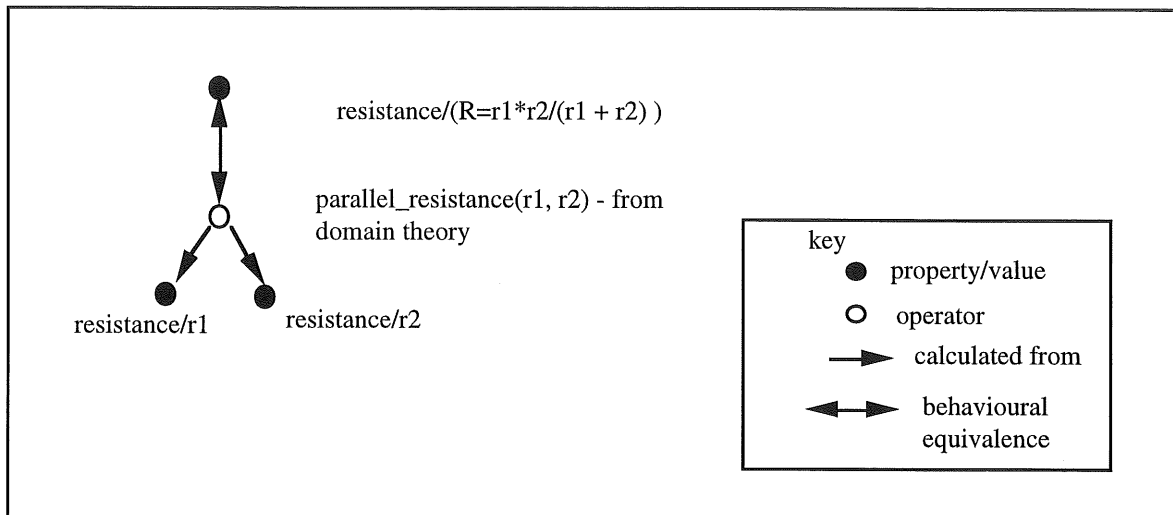


Figure 2.3: Calculation of property value from components.

3. Black box system models of notations

In the following study we avoided consideration of measures of models produced using different notations and restricted the domain to the notation tokens themselves.

3.1 Attributes of modelling notations and their possible measures

In order to apply model-based measurement to notations, we first identify a **class model** applicable to notations by

- identifying attributes of interest comprising their entity sets and associating each of these with a property variable;
- acknowledging the presence of the two mandatory relations, the co-attribute relation binding the set of property name/property variables into a coherent notation class and the co-temporal relation affirming that all the measured values exist simultaneously.

Of these, the main task is to identify the relevant attributes and suggest a suitable measurement scale for each. Both the nature of the attributes and the measurement scales are then offered for comment to the scientific community for an agreed consensus.

Most of the notations exemplified below have been chosen to cover a range of those familiar to the software engineering community and are not explained here. P/p modelling is a systems-based combination of perceptual and symbolic notations applicable to a very wide range of domains from hardware to management and quality assurance. Where the original notations have been extended by the authors or other workers, the notation is labelled 'extended'. References allow the interested reader to obtain more detailed information on each.

Attributes of modelling notations fall into four clear categories. These are examined in separate sections below and examples given.

3.1.1 Richness

A first definition is 'the number of identifiable separate entities in a key or glossary'.

A survey of the literature brought up the information contained in table 3.1, at which time several decisions had to be made about what constituted an entity. The first of the decisions to be made was on the nature of labels in diagrammatic notation. There are two opposing views. On the one hand, in a notation such as that of Data Flow Diagrams labels may be thought of as conceptual aids to link other entities to the real world or as pointers to a Data Dictionary entry and thus not part of the notation itself. On the other hand, so-called labels featured in every diagrammatic notation and could not be ignored. We thus added a token 'label' to each notation which, in a graphical notation, increased the symbolic content by a small proportion.

The numbers in table 3.1 are absolute counting numbers and as such may be subject to all the operations of that scale including common arithmetic operators such as addition and subtraction. The only care to be taken in their use is to obtain agreement on exactly which entities are to be counted.

This somewhat crude measure will need further refining to take account of the amount of information each token contains. Two examples should suffice:

a directed arc in graph theory contains more information than a simple arc;
the override token in Z contains more information than a 'decoration'.

This aspect is considered in Section 5, from which it may be possible to aggregate more precise measures.

However 'richness' is defined, the resulting attribute of the modelling formalism will have a single number for its measure. (An equivalent number will be derivable for the models themselves but this is not our concern here.)

Notation	Number of entries in key or glossary	Comments
DFD	4	FER92 p 634 ²
extended DFD	8	FER92 p 635
Structure Charts	5	FER92 p 422
ERD- Chen	5	
ERD- extended	10	FER92 p
Petrinets	4	
P/p modelling		
State Transition diagrams	4	FER92 p635
State Charts	unavailable	
Z	76 + 1	SPI87 page 150 and 1 diagrammatic schema layout
VDM	82	BIC94
CSP	104	HOA85
1st order Predicate logic	12	TUR87 page233 Var, Const, Pr and Fr counted as 1 each
Prolog	(non-operational as 1st order logic)	
Story Boarding	2	
Natural Language	8	(english) common (major) parts of speech ³
Tables	4	entity, key, foreign key, attribute
Data Dictionary	8	7 shown in YOU87 p 191 + legal character string

Table 3.1: Number of entities in notation

²As many as possible taken from same source in interests of comparability

³ {adj, adv, art, conj, noun, prep, pron, verb }

3.1.2 Perceptual/ symbolic attribute classification

In [FIT81] the authors discuss the division of modelling presentations into *perceptual* and *symbolic*.⁴ In general, perceptual presentations are those in which we perceive meaning directly without having to reason about them; they may appeal to any of the senses: colour codes (for cables), auditory codes (the telephone system) and *diagrammatic*. Only the last of these counts as a notation and needs concern us here.

In practice both types of notation, symbolic and perceptual, frequently mix in the same model. Graphs are diagrammatic but often contain labels (symbolic entities) for the various nodes. Symbolic notation is frequently laid out to show certain aspects perceptually: formatting of program code and introduction of 'white space' to aid comprehension, Z's use of schema, (spatial layouts which differentiate between declarations and axioms).

Diagrammatic and symbolic notations each have advantages and disadvantages. Models in diagrammatic notation use graph theory and two-dimensional topology, embodying ideas such as connectedness and inclusiveness, which most people understand. However, without elaborate 'extras' the amount of information they can contain is restricted and, while they may impose quality checks, they have either no or only a rudimentary deductive mechanism to apply reasoning to the model. Models in symbolic notations are able to hold much more information and can be tested for such properties as consistency and completeness. They are mostly beyond the understanding of people not specially trained in the individual methods.

Table 3.2 summarises, rather crudely, the perceptual/symbolic classification for some common modelling notations, with a further subdivision into diagrammatic-connected and diagrammatic-inclusive attributes. The first, diagrammatic connected, is graph-based, with nodes and arcs while inclusive is topological, such as used in Venn diagrams. The table shows a nominal measure (labelling) for each of the three attributes where we write yes or no as a shorthand for the column headings.

Combining the three gives a derived property, which takes one of the eight possible combinations but is still be on a nominal scale. The nominal scale is the least useful of the five basic measurement scales, allowing only equivalence to be tested.

To attain a measure 'ease of understanding' means imposing a *subjective* ordering of the eight values. This gives a utility measure on the ordinal scale in which it becomes possible to introduce the operator 'has_more_of' that particular attribute to make pairwise comparisons, where for instance rank DFD \prec VDM or use the ordering in further and more complex utility concepts. There is, however, a danger in assigning numerical values in such an ordering, the unwary could assume relationships which were not justified - relationships such as that of 'notation A is twice as hard to understand as notation B'.

⁴ Since the word 'symbolic' is used for type of presentation, those notation objects usually called 'symbols' are now referred to as 'tokens'.

	diagrammatic		symbolic	comments
	connected	inclusion		
DFD	yes	no	no	
extended DFD	yes	no	no	
STRUCTURE CHARTS	yes	no	no	
ERD- Chen	yes	no	no	
Petrinets	yes	yes	no	tokens with nodes
ERD- later	yes	no	no	
P/p modelling	yes	no	yes	
State Transition diagrams	yes	no	no	
State Charts	yes	yes		mainly inclusion for concurrency
Z	no	yes	yes	elements within frames
VDM	no	no	yes	
CSP	no	no	yes	
Predicate logic	no	no	yes	
Prolog	no	no	yes	
Story Board	yes	no		much used in industry
Natural Language	no	no	yes	for comparison

Table 3.2: Perceptual/symbolic classification

As mentioned above, any notation is unlikely to be completely diagrammatic or completely symbolic. By examining the nature of the entities for each notation we obtain a clear idea of the proportion of symbolic to diagrammatic in each of them. This is illustrated in table 3.3.

Notation	Number of entities in key or glossary	graphic entities	symbolic entities
DFD	4	4	1
extended DFD	8	8	1
Structure Charts	5	5	1
ERD- Chen	5	5	1
ERD- extended	10	10	1
Petrinets			1
P/p modelling	4		1
State Transition diagrams	4		1
State Charts	unavailable		several
Z	76	1	
VDM	82		
CSP	104		
1st order Predicate logic	12		
Prolog	(non-operational as 1st order logic)		
Story Boarding	2		
Natural Language	8		
Tables	4		
Data Dictionary	8		

(i) labels as tokens

Table 3.3: Graphic and symbolic entities in notations

3.1.3 Formality

Formality, in the context of modelling, means having a mathematical base. For a notation to be formal we have to identify the branches of mathematics on which it relies. Common branches of mathematics forming the basis of modelling notations for software systems are: graph theory, logic, set theory, elementary topology.

Ordinal measure could be assembled by the extent of the domain of mathematics employed e.g. DFDs \ll bipartite graphs. Tables are relations, equivalent to single depth trees.

	maths.	rigour ⁵	comments
DFD	graph theory		
extended DFD	graph theory	small subset	
Structure Charts	graph theory	small subset	
ERD- Chen	graph theory	small subset	
ERD- later	graph theory	small subset	
Petrinets	graph theory	with reachability	
P/p modelling	graph theory, set theory, functions relations		
State Transition diagrams	graph theory		
State Charts	topology		restricted to 2 dimensions
Z	set theory, logic		
VDM	functions, logic		
CSP	process algebra		
Predicate logic	first order logic		
Prolog	Horn clause logic		
Tables	Relational algebra		
Data Dictionary	convention		

Table 3.4: Mathematical bases of notations

3.1.4 Coverage

In this section we identify an attribute which represents the notation to represent data and processes both as atomic units and as structures and to model time explicitly. These are referred to as *facets*.

We identify eight facets of models which may be portrayed by a modelling notation and examine the ability of each notation to represent the various facets.

In order to model time explicitly, two conditions are necessary:

- the notation should have some means of expressing time units
- no cycles or feed-back should be allowed since time does not flow backwards.

⁵ How much does it enforce conformance, can misuse be prevented - ordinal scale? this is formality. eg data dictionaries as less formal than relational tables.

	Data		Processes		Time
	atomic	structure	atomic	structure	
DFD	y	n	y	y	n
extended DFD	y	n	y	y	n
Structure Charts	y	n	y	y	n
ERD- Chen	y	n	y	y	n
ERD- extended	y	n	y	y	n
Petrinets	y	n	y	n	n
P/p modelling	y	y	y	y	y
State Transition diagrams	y	n	y	n	n
State Charts	y	n	y	n	n
Z	y	y	y	y	n
VDM	y	y	y	y	n
CSP	y	y	y	y	n
Predicate logic	y	y	y	y	n
Prolog	y	y	y	y	n
Story Board	?(i)	?	?	?	n
Natural Language	y(ii)	y	y	y	yes
Tables	y	y	y	y	
Data Dictionary	y	y	y	y	no

- (i) no pattern emerged.
(ii) but with possibility of ambiguity

Table 3.5: Facets of interest shown by notations

After agreement on what facets should be included, a similarity matrix⁶ is prepared. Since combinations should cover as many of the facets as possible, the similarity between the notations should be a minimum, therefore entries with a similarity closest to zero in the table are chosen.

Table 3.5 has been composed using absolute truth values. If the table is expanded to allow fractional truth values such as 'partial' and 'undecided', then care should be taken since the two are not equivalent.

Modifiability

The notion of modifiability

Modifiability in specification relates to the ability to change part of a specification autonomously without undue and unforeseen consequences for the whole. Quantitatively one would also like to calculate the effects of the change on the important attributes of the system being specified.

Modifiability therefore can only be treated rigorously in respect of specific attributes, those calculable from the structure. The first task is to consider which structures lend themselves to modification, again using the notion of parts.

Modifiable structures

A structure of type S will be considered modifiable if there exists an operator op such that (S, op) forms a group. In other words, if s_1 and s_2 are members of the set S then $(s_1 op s_2)$ is also a member of the group.

⁶ For a brief introduction see FER92 p132

Examples:

1. If S is the set of strings, then the operator is 'concatenate' such that if s_1 and s_2 are strings then s_1 concatenate s_2 is also a string.
2. If T is the set of trees and the operator is 'with_nested' then if t_1 and t_2 are trees then t_1 with_nested t_2 is also a tree.

We can judge whether a notation has this attribute by looking at the structures formed by the notations and examining whether they allow the attributes of the whole structure to be calculated from the parts. If the answer is 'yes' then any part may be modified and the new attribute values calculated without prejudicing the coherence of the whole.

A specification expressed as Z schemas is modifiable down to the individual schemas. Several operators exist such as schema inclusion, schema composition and piping which allow for the modification of constituent schemas.

A data flow diagram is modifiable only by using the level expansion operation provided that inputs and outputs to and from each unit are preserved.

Data dictionaries and tables (as relations) are modifiable by the operations of relational algebra: join, project, select, to produce further tables.

Entity relation diagrams are not modifiable.

Prototyping

Any notation which is a programming language can be used for prototyping. The only notation examined here which embodies the ability to prototype is Prolog. However many of the others can be translated fairly easily into Prolog some instances being exemplified in [KAP94]. For example, Ada is used as a program design language and designs using the Ada notation when expanded could be used as prototypes. However, as a general rule, the translation can hold no more information than the notation itself.

3.2 The relation set of the black box system model

The relation set contains as a minimum, the two mandatory relations, the co-attribute and the co-temporal relations. The co-attribute relation, it will be remembered, affirms that all the attributes (and hence the property/value pairs) belong to the same notation. The co-temporal relation, the time stamp, refers to the date at which the notation was defined. This gives a way of version check.

Other relations between the attributes may emerge according to the attributes included and their definitions. If attributes are related in a measurable way then this indicates redundancy in the entity set. Redundancy is quite often desirable both as a quality check and to aid understanding, but when it occurs it should be made explicit.

3.3 Constructing a black box class model for notations

A class model for a notation referent may be constructed by identifying the attributes and relations, assigning the property variables to the attributes and leaving them uninstantiated. The co-attribute will not be affected but, until the individual notation and the date of its definition are known, the time stamp too will remain a variable.

Using the attributes so far investigated, the class model for the notations is:

$$S_{\text{notn}} = (\{\text{richness/E, perceptual-symbolic/P, formality/F, coverage/C, modifiability/M}\} \{r_0, t\})$$

4 Structural models of notations

Structural models were introduced in Section 2.3. This model is now examined with reference to the Data Flow Diagram notation. The parts of a notation are the tokens it uses and these will form the entity set of the systems model (expression (2)) and are shown in figure 4.1.

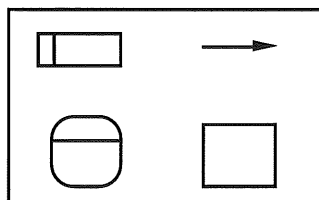


Figure 4.1: Tokens of DFDs (Gane and Sarson version)

The relation set must still contain the mandatory co-attribute relation and this is shown in the structure of figure 4.2. This relation is by no means sufficient to indicate how to use the notation. To provide a guide to this we need to add further relations, relations such as (here all binary): 'can_join/2', 'can_precede/2' and any others required, which together form the domain theory of the notation. Simple redundancy can be illustrated here by the inclusion of an extra relation 'can_follow/2', which expresses the same notion as can_precede/2'.

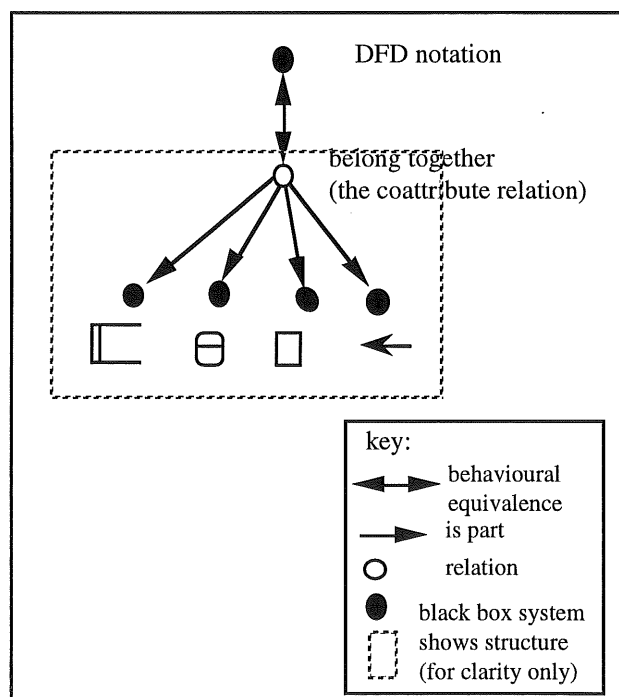


Figure 4.2: Structural digraph of DFD

A two-level structural diagram incorporating two relations is illustrated in figure 4.3.

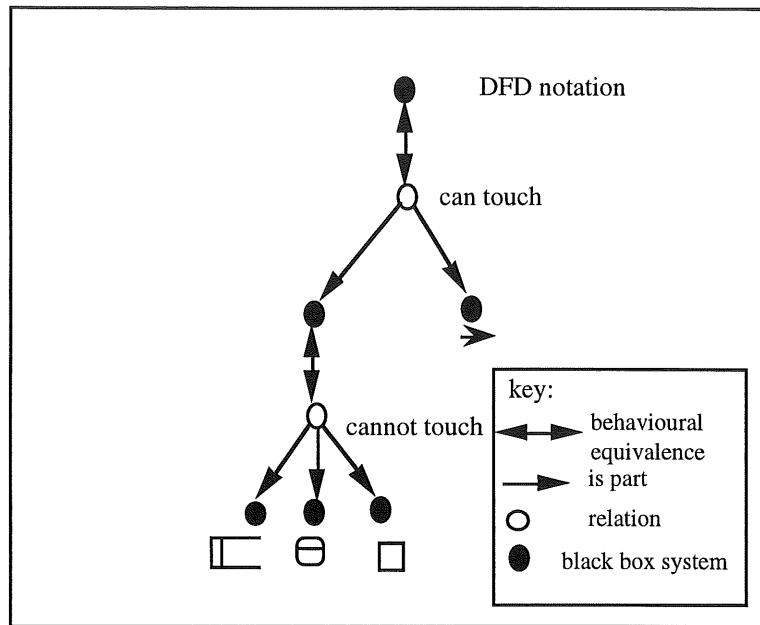


Figure 4.3: A multilevel version of DFD notation structure

Figure 4.3, in effect, partitions the set of tokens. Measures could be developed showing the cardinality of the permissible ternary combinations of tokens but this is outside the scope of the present paper.

5 Scalability

The referent shared by the black box and structural system models and indicated by the double headed arrow of the figures in Section 4 imply that, knowing any attribute of the simplest black box components and the relations forming the structure should lead to a calculated value of the attribute for the whole structure. This is known as scalability and is widely used in an engineering environment. This section examines the applicability of scaling to two cases, upward scaling for the calculation of an attribute of a combination of two notations and the reverse, the definition of an improved 'richness' attribute.

5.1 Attributes of combined notations

This first example takes two black box system models, that of Data Flow Diagrams and that of Data Dictionaries and combines them into a structure by the use of shared labels, the labels of the DFD being defined in the Data Dictionary. We consider only the attribute 'coverage', summarised in table 3.4. The DFD notation has entries (y,n,y,y,n) and the Data Dictionary notation (y,y,y,y,n) . Combining these with a logical OR yields (y,y,y,y,n) for that single attribute, allowing DFDs to be used with greater information content. This is illustrated in figure 5.1.

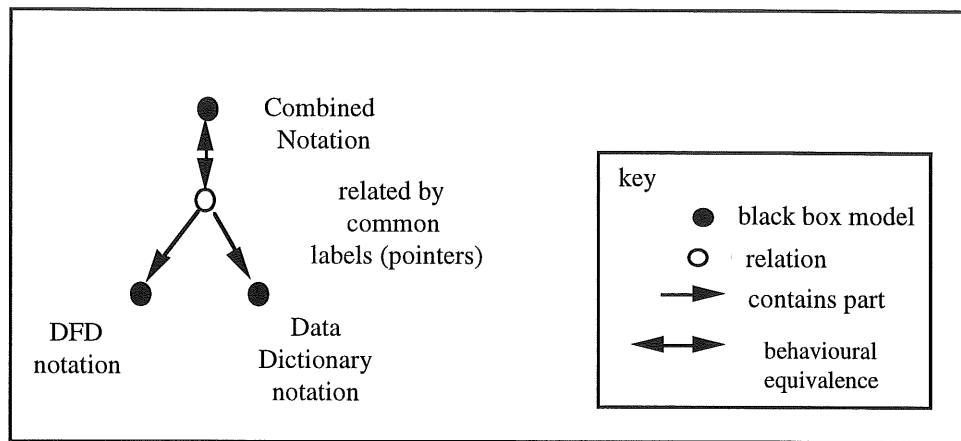


Figure 5.1 Combined DFD/Data Dictionary notation as structure

Figure 5.2 shows how one attribute may be scaled up using the structure tree. Notice how, when dealing with measures, the combining relation is shown as an evaluable function.

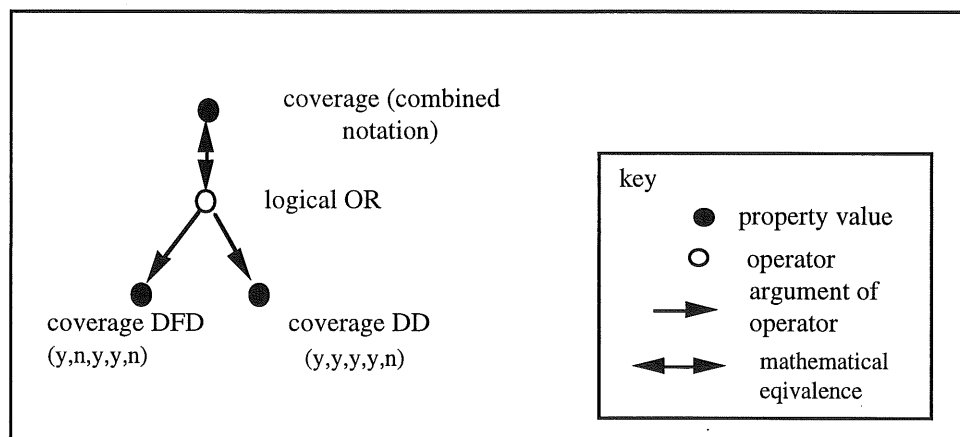


Figure 5.2: One property measure of combined DFD/Data Dictionary notation

In the cited case, as in the case of figure 2.3, the calculations can be made because of the existence of a domain theory. Without such a theory, nothing can be said about the black box attributes of a structure related to its components.

5 2 Increasing granularity and 'richness'

Scaling measures of attribute values up the structure tree is clearly of interest but scaling down, increasing the granularity by replacing the atomic black boxes as possible structures, also yields insights of value.

Consider the attribute 'richness' defined crudely in Section 3.2 as 'number of elements'. This definition ignores completely any conceptual content of individual elements. An example is provided by the override element in Z (I) This contains much more intellectual 'baggage' than a simple + sign and thus may be considered a more significant element or, to revert to the language of measurement, should be worthy of a higher value than a crude count of '1'. Assigning worth by intuition is unscientific. At best it can only produce an ordinal scale and measures on an ordinal

scale cannot be composed other than by an independent test to determine position in the ranking.

Figure 5.4 contrasts the original measure of richness (on the left) with a better version (on the right). It involves an explicit measuring process for each entity in the element set of a notation, a process which will, in some way, reflect how rich it is in meaning when mapped into the real world.

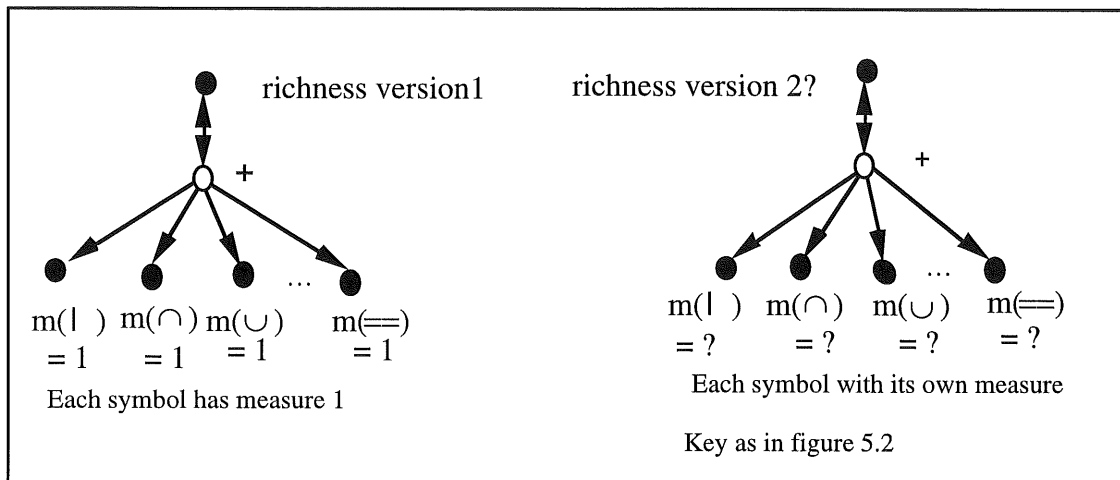


Figure 5.3: A better 'richness' measure

A fully worked out modelling and measurement scheme for assigning such measures has been created for specification languages and brief examples appear in KAP94. The method involves expressing and defining the element as complete Prolog procedures and measuring the data of the definition in terms of a six-tuple (A, V, I, C, S) where A is the number of arguments, V is the number of independent variables, I the number of identities, S the number of structures and D the maximum depth of structure. Where alternate definitions are possible, the minimum six-tuple is taken.

A tool exists for the automatic production of such six-tuples.

6. Indirect and utility measures

Until such time as definitions are agreed for direct measures, it would be premature to look at indirect and utility measures. Nevertheless some lines of investigation were suggested during the current work. They are included here as pointers.

6.1 Understandability

Diagrammatic notations are usually considered to be more comprehensible than symbolic ones. Understandability could then be a function of perceptual/symbolic content:

$$\text{understandability} = f(\text{perceptual}, \text{symbolic}).$$

Table 3.3 suggests such a function might be

$$\frac{\text{count(perceptual)} - \text{count(symbolic)}}{\text{count(perceptual)} + \text{count(symbolic)}}$$

giving a range from +1 to -1.

6.2 Compatibility

Compatibility is a binary function showing how the coverage of notations can be extended by a judicious combination.

A compatibility matrix is built from the information in table 3.4 extended to include such facets as 'shows sequence' and others.

As the matrix implies, compatibility between notations is a pairwise comparison. There are various considerations:

1. Does the notation in one conflict with the notation in the other? Structural decomposition of processes does not integrate with structural decomposition of data.
2. Does the notation of one complement the other? If yes, then parallel specification of structures and modules would be helpful, allowing modules to be specified and then assembled into a whole.
3. If one notation holds less information than the other then is it an abstraction of the other? Does the simpler notation allow preliminary decomposition before a fuller specification is attempted?

7. Discussion

The task undertaken was a preliminary investigation of measurement of notations in the context of specification of multimedia for educational systems. It turns out that the restriction to one type of specification was not helpful and this paper presents a foundation for measurement of any specification notation.

At present developers, in need of a modelling notation, are faced with an ever increasing choice, too varied to try out in practice, all valiantly attempting to keep up with the accelerating development of technology and communication methods. Such rapid change means that empirical testing is neither feasible nor sensible, therefore, this and the companion report from the M3 project [BRI96a] have adopted a theoretical approach. Taking our cue from engineering proper we have sought measures which may, in the future, lead to the development of a satisfactory theory on which to build and validate such measures as will be of real use to developers.

We are conscious that there is a lack of any consideration of specification of the user interface. Here we await developments in the field of cognitive psychology and education.

Appendix A1 summarises the 5 basic measurement scales, some of which are mentioned in the text.

Appendix A2 shows how measurement can be applied to tree structures such as that of figure 5.3 and expansions of it.

8. References

- BIC94 Bicarregue J, Fitzgerald J, Lindsay P, Moore R, Ritchie B, *Proof in VDM*, Springer-Verlag 1994
- BRI96a C.Britton, S.Jones, M.Myers and M.Sharif *Notations for modelling multimedia systems* Technical Report No. 256, Faculty of Information Sciences, University of Hertfordshire, 1996
- COH86 Cohen B, Harwood W. Jackson M, *The Specification of Complex Systems*, Addison Wesley, 1986.
- FER92 Fertuck L, *Systems Analysis and Design*, WCB, 1992
- FIT81 Fitter M, Green T R G, *When do Diagrams Make Good Computer Languages?* in *Computing Skills and the User Interface*, Eds. Alty J and Coombs M, Academic Press, 1981.
- GRE89 Green T R G: *Cognitive Dimensions of Notations*, in *People and Computers*, eds: Sutcliffe and Macaulay, COP 1989.
- HOA895 Hoare C A R: *Communicating Sequential Processes*, Prentice Hall, 1985
- KAP94 Kaposi A, Myers M, *Systems Models and Measures*, Springer Verlag 1994
- KAP96 Kaposi A, Myers M, *Product/process Modelling - an overview* Kaposi Associates 1996
- PRE94 R S Pressman, *Software Engineering, a practioner's approach*, (European Edition), McGraw Hill 1994
- SPI87 Spivey J M, *The Z Notation A Reference Manual*, Prentice Hall 1987
- WIE95 Wieringa R J: *Combining Static and Dynamic Modelling Methods*, *The Computer Journal* Vol 38, No. 1, 1995
- YOU89 Yourdon E: *Modern Structured Analysis*, Prentice Hall, 1989
- ZAD75 Zadeh L, *Fuzzy Sets and their Applications* in *Fuzzy Sets and their Applications to Cognitive and Decision Processes*, eds: Zadeh L, King-Sun Fu, Kokichi Tanaka, Masamichi Shimura, Academic Press, 1975.

Appendices

A1. Measurement scales

nominal

chosen from a pre declared set.

Examples: {black, white, red, blue, yellow, green}
 {capricorn, etc.}

There is no ordering implicit in the label measures. Defining relation (allowable for comparison of a similar attribute of two entities is) *equivalence*

ordinal

impose ordering on the labels, depending on some attribute such as height, weight.

Examples: {good,better,best}
 {a,b,c,d,e} in the grading of eggs.

Defining relations are *equivalence, greater_than, less than.*

interval

values on an interval scale with an arbitrary zero.

Examples: height above sea level, degrees celsius.

Defining relations are *equivalence, greater_than, less than.,relative scale values*

ratio

values on an absolute scale - i.e. possessing an absolute zero.

Examples: degrees celsius, length, mass.

Defining relations are *equivalence, greater_than, less than., relative scale values, ratio between scale values.*

absolute

counting scale

Examples: 5 (oranges,) 12 (months)

Defining relations are *equivalence, greater_than, less than.,relative scale values, ratio between scale values, absolute scale values.*

A2 An example of measurement of structure

Measurement of trees (acyclic digraphs with single entry node).

All measures on trees may be derived from two sets:

1. the dispersion set **P₁** - the set of out-degrees of all nodes;
2. the path-length set **P₂** - the set of all path lengths.

An example is provided in figure 1

Absolute measures on the tree:

P₃ (node count) = the cardinality of set **P₁**

P₄ (arc count) = **P₃** - 1

P₅ (leaf count) = the cardinality of set **P₂**

P₆ (longest path length) = max(**P₂**)

P₇ (maximum fan-out) = max(**P₁**)

P₈ (cover) = sum of elements of **P₂**

Ratio measures can be defined on these if necessary, giving depth, width, leafiness.

In contrast to the previous measures, the ratio measures will not be integers but will also be dimensionless (See Systems, models and Measures).

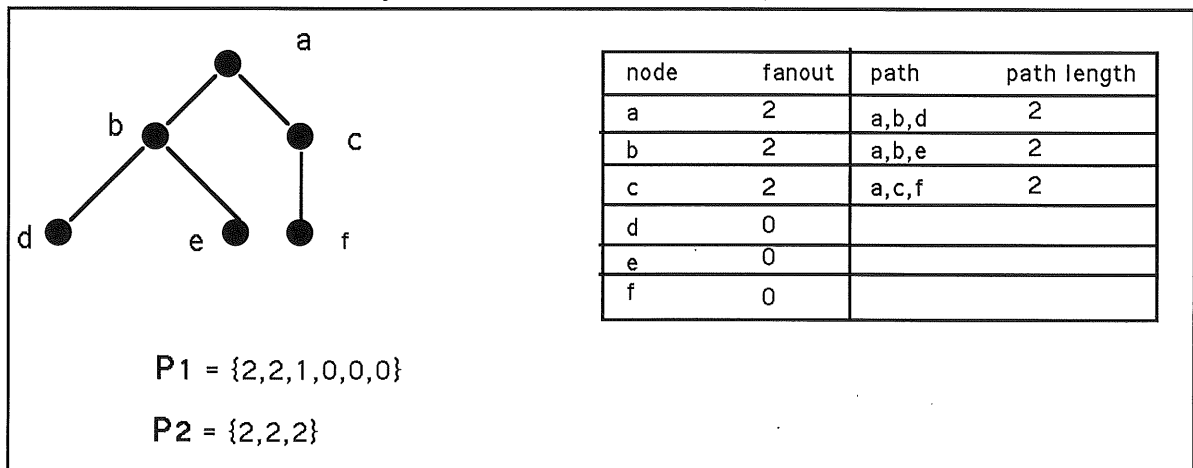


Figure A1: Basic tree measures - an example

V2. Extension of measures to cyclic graphs.

Model of a multi-media system will be single entry and can therefore be represented as a tree structure. Changing to a reachability graph will add arcs, and create potential cycles. We will then not have a finite path length set.

The dispersion set will be different but finite. Any measures based on **P₁** above are therefore still valid. These are **P₃**, **P₄**, **P₇**.

In addition, we can derive ratio measures from these three taken before and after the addition of the extra arcs. To assess the 'meaning' of these measures would require investigation of a fair number of samples (more 'further work')

There is another option and that is to use the *minimal spanning tree*

Definitions: a *spanning tree* of a graph G is a tree such that it covers all the nodes of G . A *minimal spanning tree* of G is such that the sum of the lengths of the edges is minimal among all the spanning trees.

The full set of tree measures then become applicable.

⁷ as functions over data