CHAPTER FOUR – KNOWLEDGE REPRESENTATION

This part of the manual describes the process of knowledge representation. Knowledge representation comprises;

- the process of abstracting knowledge
- the process of formal representation
- the process of formal term specification

4.1 ABSTRACTING KNOWLEDGE

4.1.1. BASIC CONCEPTS

The knowledge given by an informant is made up of a combination of basic units of knowledge. These units will be structured according to the context in which they are articulated. In the creation of a knowledge base we are more interested in the individual units used to construct that combination, than in the combination itself. We also need to understand the ways in which these units can be linked.

It is more important to 'disaggregate' and record the individual units of knowledge that make up a particular statement, than to record the original combination. The units can be recorded along with enough associated information to allow them to be re-combined with others to explain aspects of the behaviour of a system.

The first step in the process of knowledge representation therefore is to abstract these basic units from the knowledge articulated by the informant. These basic units are referred to here as 'unitary statements'.

4.1.2 UNITARY STATEMENTS AS BASIC UNITS OF KNOWLEDGE

The term 'unitary statements' is used here to refer to the smallest useful units of knowledge. A unit of knowledge is useful if it can be used in combination with other knowledge in reasoning.

Unitary statements express assertions. A statement is not the same as a sentence because different sentences can be used to express the same statement. Unitary statements contain knowledge that is useful without reference to other unitary statements but which cannot be broken down into further unitary statements. So:

> *"The biomass production of a plant is proportional to leaf area index of that plant."*

is a unitary statement.

By contrast:

> *"Rainfall is low."*

is not a unitary statement, because it does not contain enough information to be used in reasoning; it does not refer to any system, any time or any place.

> *"Clover is eaten by sheep and goats."*

is not a unitary statement, by this definition, because it can be broken down into two unitary statements:

1. *"Clover is eaten by sheep."*
2. *"Clover is eaten by goats."*

Unitary statements are divided into two categories, binary statements and attribute-value statements.

A binary statement captures a relationship between two entities, e.g.:

"Rainfall causes increased soil erosion."

Attribute-value statements describe an attribute of an entity or classes of entities, e.g.:

"Bananas are yellow."

This distinction is important in the diagrammatic representation of knowledge. Within AKT, binary statements can be represented diagrammatically, but attribute-value statements cannot.

Unitary statements can also be divided into those which refer to classes or types of entities (termed 'statements of class') and those which refer to particular instances (termed 'statements of instance'):

Statement of class	*"Barley seeds germinate in seven days at 10°C."*
Statement of instance	*"Seedling 97 germinated after nine days at 10°C."*

These two forms of statements are syntactically identical but they are significantly different in utility. Explanatory ecological knowledge can be derived from statements of instance, however, statements of instance are, in themselves, data and are unlikely to constitute a useful component of a knowledge base.

4.1.3 RECORDING THE CONTEXT OF A STATEMENT

Knowledge is inescapably contextual. Disaggregation of knowledge into a set of unitary statements will cause this context to be lost. This loss of contextual understanding demands explicit information about the circumstances under which the statements apply (i.e. conditions) and the source of each statement.

4.1.3.a Conditions

Most unitary statements will have only a limited validity. Validity relates to the circumstances or conditions under which the unitary statement is held to be true and the certainty that can be placed on the unitary statement being true. Neither are adequately captured in unitary statements themselves. Appending conditional information and certainty to a statement, results in a more complete record of the knowledge articulated by an informant. However, it is extremely difficult to elicit meaningful information about the confidence informants have in the knowledge that they articulate. Furthermore, use of statements of certainty is problematic. For this reason AKT provides facilities for recording conditions, e.g.:

"Soil erosion is severe IF
the slope is greater than 20° AND
rainfall is over 1000 mm per annum AND
vegetation cover is thin."

but it does not give any indication of the confidence the informant, or others, have in this particular piece of knowledge. Although attempts have been made to capture the degree of confidence in a statement, it was found that informants were uncomfortable in providing statements of confidence, in part because it appeared to question their veracity.

4.1.3.b Source

One important piece of contextual information for the interpretation and use of knowledge is the source of that knowledge. Where a user of knowledge is familiar with or aware of the informants/references, this will inform his/her use of that knowledge. For this reason knowledge is tagged according to source. This also facilitates the assessment of the internal consistency of knowledge from a particular source and the distribution of knowledge between sources.

There is an ethical need to enter the source of knowledge, otherwise one may lay oneself open to accusations of extractivism. By recording the source of knowledge, it is also possible to identify whose knowledge is being used at any time so that if there is any value attached to that knowledge it can be attributed to the original source. However, in general the methodology is designed to record how a community understands the resources on which its livelihood is based, and the purpose is not to seek out specialist (valuable) knowledge, but to collect the knowledge of a community, rather than an individual.

4.2 FORMAL REPRESENTATION

4.2.1 INTRODUCTION

Because of the ambiguity and complexity inherent in natural language, accurate interpretation of unitary statements in unrestricted natural language may often be difficult. Natural language is extremely robust in its use and interpretation; it can contain a great deal of ambiguity and imprecision, yet still serve a useful function in communication. Meaning in natural language is often context specific and, therefore, flexible. However, it cannot be assumed that the implicit contextual meaning of a unitary statement will still be understood by users once it has been included in a knowledge base. Furthermore, automated reasoning techniques cannot cope with flexibility of meaning according to context. As a result, the second stage in the creation of a knowledge base is to create a version of the unitary statement that conforms to a formal syntax. This process is known as formal representation.

The process of formal representation results in statements that have a fixed syntax and can therefore be combined using inference mechanisms in automated reasoning (see Chapter 9). The process also provides a means of evaluating the knowledge already elicited, rationalising terminology and identifying ambiguity, inconsistency and so on (see Chapter 9).

Formal statements comprise a set of terms linked by, and ordered in relation to, other specific terms that form part of the formal language. This linkage and ordering provides information on the way in which the elements of the statement are related and therefore enables a semantic interpretation of the syntax of the statement.

The formal grammar that provides the basis for formal representation is stated in Table 4.1. A summary explanation of the grammar is provided in this section and a more detailed exploration of its application follows.

Table 4.1 *The definite clause grammar. Terms in bold are reserved terms in the grammar (i.e. words reserved for use by the system); terms starting with a capital letter are variables; ⇒ means 'can take the form of'.*

FormalSentence ⇒ Statement **if** FormalConditions
FormalSentence ⇒ Statement
Statement ⇒ Cause **Causes** Effect where Causes is an element of the set:{**causes1way**,**causes2way**}
Statement ⇒ AttributeStatement
Statement ⇒ **not**(AttributeStatement)
Statement ⇒ **link**(influence,Thing,Thing)
Statement ⇒ **link**(Link,Object,Object)
Statement ⇒ **link**(Link,ProcessBit,ProcessBit)
Statement ⇒ **link**(Link,ProcessBit,Object)
Statement ⇒ **comparison**(Attribute,Object,Comparison,Object)
FormalConditions ⇒ FormalConditions **and** FormalConditions
FormalConditions ⇒ FormalConditions **or** FormalConditions
FormalConditions ⇒ Statement
FormalConditions ⇒ ActionBit
FormalConditions ⇒ ProcessBit
AttributeStatement ⇒ **att_value**(Object,Attribute,Value)
AttributeStatement ⇒ **att_value**(ProcessBit,Attribute,Value)
AttributeStatement ⇒ **att_value**(ActionBit,Attribute,Value)
Cause ⇒ AttributeStatement
Cause ⇒ ProcessBit
Cause ⇒ ActionBit
Cause ⇒ Object
Cause ⇒ **not**(Cause)
ActionBit ⇒ **action**(Action,Object,Object)
ActionBit ⇒ **action**(Action,Object)
Effect ⇒ AttributeStatement
Effect ⇒ ProcessBit
Effect ⇒ ActionBit
Effect ⇒ **not**(Effect)
Process_bit ⇒ **process**(Process)
Process_bit ⇒ **process**(Object,Process)
Process_bit ⇒ **process**(Object,Process,Object)
Thing ⇒ Object
Thing ⇒ ProcessBit
Attribute ⇒ atom
Process ⇒ atom
Link ⇒ atom
Object ⇒ atom
Object ⇒ **part**(Object,Object)
Action ⇒ atom
Comparison ⇒ Atom where Atom is an element of the set:{**greater_than**, **less_than,** **same_as**, **different_from**}
Value ⇒ Atom Where Atom is an element of the set:{**increase**, **decrease**, **change**, **no_change**}
Value ⇒ Atom
Value ⇒ Number Where Number is either a floating point number or an integer
Value ⇒ **range**(Value,Value)

4.2.2 THE ELEMENTS OF FORMAL REPRESENTATION

There are three basic elements to formal representation: **objects, processes** and **actions.** Objects, processes and actions can be considered to be the fundamental elements around which the formal statement is structured.

Objects are normally physical items in the real world, like 'trees', and 'crops', but may be conceptual, for example, 'niche' or 'wet season'. The name of objects in a formal statement are represented by atoms, so, *cow*, *tree* and *hill* are objects represented as atoms. However, objects in formal statements are usually members of a class of atoms (for example *cows*, *trees* and *hills*). Atoms are represented in lower case only. Where an atom in a formalised statement consists of two or more words, these should be joined with an underscore (e.g. windfall_apples). The exception to this rule is when the object name in a formal statement begins with a capital letter (e.g. in a proper or Latin name). In this case the name should be enclosed between single inverted commas (e.g. 'Nepal').

Processes (or events) describe changes or fluxes in the real world, for example the process of soil erosion describes the loss of soil, and the process of germination describes the change in a seed from dormancy to active growth. Processes, like objects, are named and are also represented by atoms. In some circumstances a process is not associated with any particular objects in the statement;

e.g. *'rainfall'*;

alternatively, it may be related to one identified object

e.g. *'erosion'* is a process related to the object *'soil'*

or may provide linkage between two objects

e.g. *'infestation'* is a process which links the object *'pests'* with the object *'crop'*; *'uptake'* is the process that links the object *'plant'* with the object *'water'*

An **action** is similar to a process but is initiated by man; for example, it incorporates all deliberate actions carried out by managers of an agroforestry practice and is always related to either one object or two,

e.g. *'ploughing'* is the action related to the object *'field'* , and
'application' is the action linking the object *'pesticide'* to the object *'crop'*

Statements are formed by combining these three elements with **attributes**, **values**, **(user defined) links** and a set of special 'reserved' terms used in the formal language (see below 4.2.2.a).

An **attribute** describes an object, process or action and is generally measurable. Thus *height*, *rate*, *colour*, *frequency*, *gradient* are all attributes. An attribute is represented as an atom.

A **value** is always attached to an attribute, and describes that attribute. These values can be in units, for example, *5 kg*, *20 hectares*, *40_kg_per_ha_per_year*, *3_months_7_months*, or they can be descriptive values, such as *tall*, *rapid*, *yellow*, *regular*, *steep*. A value can be represented as an atom, a number or as a range.

A **(user defined) link** is a term selected to link two objects, two processes or a process and an object, when these cannot be linked using the reserved terms (see below 4.2.2.a). Thus in the statement 'cows eat grass', *eat* is the user defined link, in the statement 'fruit bats pollinate *Parkia biglobosa*' *pollinate* is the (user defined) link. A (user defined) link is represented as an atom.

These above ingredients provide the basis for formal representation. Using different combinations of these, there are four types of statement that can be formed:

- Attribute Value statements
- Causal statements
- Comparison statements

- (User defined) Link statements

(See 4.2.4 below)

4.2.2.a Reserved terms

- The term **comparison** enables the comparison of the value of an attribute for two objects[1].
- The term **causes** allows a causal relationship between an attribute-value statement, object, process or action and another attribute-value statement, process or action to be captured.
- The term **if** allows conditionality to be captured whereas the terms **and** and **or** allow multiple conditions to be specified for a unitary statement.
- The term **part** allows a particular part of a specified object (for example the roots of wheat) to be represented.
- The term **not** can be used to capture negation (e.g. pigs do not eat grass).
- The term **link** allows relationships between objects or processes other than causal or comparative links to be captured (e.g. cows eat grass).

Some elements of formal statements may be represented by atoms that have special meanings and are used in particular contexts. For example, a value may be represented by the special terms **increase**, **decrease**, **change**, **no_change** and **range**. Comparison types can be expressed by one of the following terms; **greater_than**, **less_than**, **same_as**, or **different_from**. Finally, under some circumstances, a link may be represented by the term **influence**.

The grammar was designed with a particular emphasis on causal, comparison and attribute-value statements. However, management actions (i.e. deliberate actions carried out by the managers of an agroforestry practice) are distinguished from ecological processes and can be represented where a statement captures the impact of the management action on the ecology of the practice.

4.2.3 FORMAL REPRESENTATION OF SINGLE UNITARY STATEMENTS

A detailed description of the application of the formal grammar and justification for its structure is best achieved through example.

The following guidelines are divided into four parts:

a) re-evaluation of the single unitary statement;
b) identification of fundamental elements;
c) identification of statement structural type; and
d) creation of the formal statement.

4.2.3.a Re-evaluation of the unitary statement

The first step in the process of formal representation is to reassess the individual unitary statement. For each new unitary statement one should ask oneself the following questions:

i. Is the statement clear?
ii. Is it sensible?
iii. Is it unambiguous?
iv. Is it complete?

[1] To compare the process of two objects (such as growth, or germination), then the process must be made into an attribute of the object, e.g. rate_of_growth or rate_of_germination.

v. Is it a genuinely unitary statement (and not a compound statement)?

If the answer to all five questions is not 'Yes' then the unitary statement should be reconsidered. In particular, statements are frequently found to be compound, so that it is necessary to break them down into individual unitary statements.

Formal representation is justified by the automated reasoning (Chapter 6) that it makes possible. However, the rigorous consideration of the meaning of each unitary statement that is demanded by the process of formal representation is of significant value in its own right.

4.2.3.b Identification of the elements in the statement

The second step in formal representation is to identify the objects (and parts of objects), processes, actions, links, attributes and values in the statement.

This step is important because knowledge must be explicitly and unambiguously stated in creating formal statements that provide a robust resource for automated reasoning. By contrast, natural language tends to contain implicit elements. A distinction can be drawn between (a) an implication resulting from inadequately abstracted statements (where domain-specific knowledge may be needed to achieve more explicit representation) and (b) an implication that the user of the system might understand but which is unacceptable for formal representation.

The fact that the statement:

Khasru causes sickness in cattle

means that cattle become sick if they eat Khasru, is an example of (a) because the user needs to know that Khasru is eaten.

By contrast, and as an example of (b), it might be assumed that any user of the system can interpret the statement:

Utis is a tall tree

as being about the height of the tree, while formal representation demands that this is explicitly stated,

i.e. *the height of the Utis is tall*

Attributes are often implicit. For example, in the statement:

Loam is very fertile

'loam' (a type of soil) is an object, 'very fertile' is a value but it is not immediately clear of what this value is a measure (i.e. the attribute to which it refers). However, the attribute must be identified in formal representation. 'Soil fertility' might be suitable in this case.

By contrast, objects, processes and values are implicit only occasionally, usually only where the unitary statement is inadequate (as with the implicit process in the first example above) or where representing statements that stretch the use of the grammar. For example, the statement:

Rainfall is maximum in January

does not contain an explicitly stated object. Rainfall is taken to be a process. In other circumstances, it might be viewed as an object but 'maximum in January' implies that the attribute is a rate rather than a volume. Rates can only be associated with processes and actions. So, 'maximum' is a value for the attribute 'rate' of the process 'rainfall'. 'January' is a value for an attribute time, or maybe time of year. The attribute 'time' refers to the time at which the event occurs.

The statement:

Soil erosion reduces soil fertility

contains implicit values. Fertility is an attribute of an object 'soil' while soil erosion is a process with an implicit attribute 'rate'; both attributes have implicit values, in this case 'decrease' and 'increase' respectively. Therefore the statement expressed explicitly would read:

an increase in the rate of soil erosion causes an decrease in soil fertility

'Part' relationships between objects are also identified at this stage. For example, in the statement:

Siris has a light crown

Siris (a tree species) and crown are objects, however, they are further related in that the crown is a particular part of the object Siris. The identification of 'part of' relationships is similar to the identification of 'type of' relationships between objects in the object hierarchy and can similarly be used in reasoning.

4.2.3.c Identification of the statement structural type

The third step in the process of formal representation is to identify which type of formal statement best captures the meaning of the unitary statement. The identification of structural type, i.e. binary statements or attribute value statements, was introduced in **4.1.2**. However, the process can be more demanding during formal representation because some statements which cannot be represented directly in the formal grammar (for example temporal *statements) can be represented using the other, existing, structural types.*

4.2.3.d Creation of the formal statement

Formal statements may be of one of five types: a causal statement, a comparison statement, a link statement, an attribute statement or a negative attribute statement (see Table 4.1). Some unitary statements may be captured by more than one formal statement type. However, the different types have a differing utility in reasoning with the knowledge base. In general, causal statements are the most useful. Any type of statement may additionally have conditional information attached.

4.2.4 STATEMENT TYPES:

4.2.4.a Attribute Value statements

The most basic form of statement is the attribute-value statement. An attribute-value statement is descriptive, it describes an object, or process or action. An attribute-value statement for an object takes the form:

att_value(Object[2], Attribute, Value) e.g. att_value(tree, height, tall)

An attribute-value statement for a process takes the form:

att_value(Process, Attribute, Value) e.g. att_value(process(leaf, decomposition), rate, slow)

Entire formal unitary statements can be captured as attribute-value or negative attribute-value statements where the statement consists of a single object or process or action and information about the value of an attribute of that object or process:

Unitary statement	**Formal statement**
Siris has small leaves	*att_value(part('Siris', leaf), size,small)*
Sirius does not have big leaves	*not(att_value(part('Siris', leaf), size, big))*

These statements may also occur within causal statements. For example:

[2] A capital letter is used for this 'argument' of the att_value 'clause', to denote a variable

Unitary statement	**Formal statement**
The small leaf size of Siris causes a low shading effect	*att_value(part('Siris',leaf), size,small)causes1way att_value(process(shading),effect,low)*

4.2.4.b Causal statements

Given that a causal statement takes the general structure:

X causes Y

Y will usually be a change in the value of an attribute. This change can be captured by using one of the special values 'increase', 'decrease', 'change' or 'no change'. X may also be a change in the value of an attribute (again taking one of the four special values) or may be a process or an action. The grammar also allows objects to be a cause (for example, sheep cause soil creep) but this is incomplete, containing implicit information. The complete statement (for example, 'trampling by sheep causes soil creep') is always preferable, but the feature is retained in the grammar for cases in which it is not known how an object causes a change, but simply that something about its presence does.

Thus a causal statement can take one of the following forms:

Attribute statement causes attribute statement	e.g. *A decrease in stem thickness causes a decrease in stem strength.*
Process causes attribute statement	e.g. *Soil erosion causes a decrease in soil fertility*
Action[3] causes attribute statement	e.g. *Ploughing slopes causes an increase in soil erosion*
Object causes attribute statement	e.g. *Cows cause an increase in soil compaction*

Causal statements can be divided into those for which the reciprocal is also true and those for which it is not. For example, both the statements:-

a) An increase in atmospheric temperature causes an increase in germination rate[4];
and
b) A decrease in atmospheric temperature causes an decrease in germination rate

might be held to be true. As they are reciprocals of one another it is desirable to be able to enter just the one into the knowledge base and be able to infer the other.

However, the reciprocal is not always true. For example, while it may be true that:

An increase in intensity of rainfall causes an increase in surface moisture.

it is not the case that

A decrease in intensity of rainfall causes a decrease in surface moisture.

because it is not rainfall, but evaporation, drainage, etc. which dictate the rate of drying.

To capture this difference, two versions of causes are supported: **causes1way** and **causes2way**.

For these six examples, these result in the formal statements:

[3] An action should usually be attached to at least one object, *ploughing* to *field*, *harvesting* to *crop* etc. Although the grammar allows an action to stand alone, this is less useful in terms of the information it imparts and should therefore be used only sparingly.
[4] rate – percentage seed germinating within a given time

1	*att_value(stem, thickness, decrease) causes2way att_value(stem, strength, decrease)*
2	*process(soil, erosion) causes1way att_value(soil, fertility, decrease)*
3	*action(ploughing,slopes) causes1way att_value(process(soil,erosion),rate,increase)*
4	*Conifers causes1way att_value(process(soil,acidification),rate,increase)*
5	*att_value(atmosphere,temperature,increase) causes2way att_value(process(seed, germination), rate, increase)*
6	*att_value(process(rainfall),rate, increase) causes1way att_value(surface_ moisture, value, increase)*

Not all causal statements describe a change in the value of an attribute. Y may also be a process in other words, something (an attribute statement, an action, a process or an object) causes a process to take place, for example;

Object causes process *Conifers cause soil acidification*

Y may also be an action, for example;

Attribute statement causes action *An increase in pest numbers causes the application of pesticides.*

Nevertheless, we are creating ecological knowledge bases, and as actions are usually determined by a far more complex set of factors than ecological conditions alone, (a farmer's decision to apply pesticides, for example, will be dependent, not only on the increase of pest numbers, but also on economic constraints such as affordability, availability and time) and thus an action will rarely simply be an effect of an ecological state alone.

4.2.4.c Comparison statements

Comparison statements compare the relative value of a pair of objects.

The formalised comparison statement takes the following form:

comparison(Attribute, Object1, Comparison_type, Object2)

Comparison statements may be self evident, for example:

Bamboo grows faster than fruit trees.

Frequently, however, comparison is implicit, usually against an implicit 'norm'. The unitary statement:

Bans leaves decompose slowly

can be interpreted as being a comparison with the average rate of leaf decomposition. This kind of implicit comparison is, however, best captured as an attribute value statement, in this case:

att_value(process(part(bans, leaf), decomposition) rate, slow)

The only instances in which implicit comparison may best be represented as comparative statements are those in which there are clearly only two possible circumstances. So, for example, the statement:

Forests with closed canopies cast deeper shade

is a genuinely comparative statement and might be more explicitly stated as:

Unitary statement:

Forests with closed canopies cast more shade than forests with open canopies.

Formal statement:

comparison(depth_of_shade, closed_canopy_forest, greater_than, open_canopy_forest)

4.2.4.d (User defined) Link statements

This link is specified as (user defined) link in order to distinguish it from certain reserved terms that act as links between two parts of a statement. For example, the reserved terms *causes1way*, *causes2way*, *greater_than*, *less_than, same_as* link the two halves of a formal statement together. User defined links on the other hand, are terms selected by the user.

User defined link statements take the basic form:

link(link_type, Object1, Object2)

Ecological relationships such *as cows eat grass* and *bees pollinate clover* are good examples of user defined link statements, and would be expressed in formal language thus:

link(eat, cows, grass)
link(pollinate, bees, clover)

User defined link statements are also used when the knowledge cannot be expressed using any of the other three types of statement, e.g. *Tithonia diversifolia is found on unfertile ground* or *oak is used for timber*. These would be formally represented as:

link(is_found_on, 'Tithonia diversiflora', unfertile_ground
link(is_used_for,oak,timber)

The grammar includes one type of link statement, in which the link type is 'influences'. In this instance the link may be between any combination of objects and processes. Influence relationships are very closely related to causal relationships. However, in an influence relationship there is no information on what attribute of the object or process is affected or how it is changed, e.g. *Trees influence crop yield*[5]. Where there is information on the result of the influence, this should be captured as a causal statement.

4.2.4.e Representation of conditions

Statements may be conditional. Conditions in the formal language can take the form of attribute value or negative attribute value statements, link statements, causal statements and comparison statements. All these statements are as previously described and may also occur as the main part of a formal statement.

Conditions may be linked by **and** and **or**. For example:

> *Crops are prone to lodging **if** (there is a strong wind **and** crop roots are exposed) **or** (there is a strong wind **and** crop stems are weak),*

or equally:

> *Crops are prone to lodging **if** (crop stems are weak **or** crop roots are exposed) **and** (there is a strong wind).*

4.3 FORMAL TERMS SPECIFICATION

Structural manipulation of the knowledge base involves the development of consistent and useful glossaries of terms and ensuring the consistent use of those terms within unitary statements.

[5] In formal language this is expressed as: *link(influences, trees, crop_yield)*

The development of glossaries of attributes, processes, actions, values and link types is automated in AKT. Objects are automatically identified but the user is required to identify 'type of' relationships and 'part of' relationships through the creation of object hierarchies and application of the formal grammar respectively.

However, while the mechanisms for creating glossaries and hierarchies are straightforward, the management of hierarchies and glossaries is a demanding task. The consistent use of terms has a profound impact on the utility of the resulting knowledge base, particularly in relation to the use of automated reasoning tools. The consistent use of terms is facilitated by:

- minimising the number of object, attribute, process, action, value and link terms used:
- ensuring the consistent use of values for attributes and
- providing definitions for each term, such that their use is transparent and can be assessed by the knowledge base developer or other users.

The creation of object hierarchies makes further demands. Sets of objects can be hierarchically classified in many different ways for different purposes. Indeed it may not be possible to develop a single classification of a set of objects that is appropriate for all the intended uses of a knowledge base. For that reason it is possible for an object to appear in several hierarchies, although it is not possible for an object to appear more than once in the same hierarchy.

4.4 DIAGRAM BASED REPRESENTATION

4.4.1 INTRODUCTION

The representation of knowledge as a diagram provides a powerful means of helping to ensure that the knowledge set that is represented is comprehensive and coherent and therefore useful.

The diagramming approach to representation is not constrained by linearity. A set of unitary statements may be legitimately explored from any point in any direction. As a result, diagrams provide a more succinct representation of knowledge than textual approaches.

A further advantage of diagrammatic representation is that statements entered through the diagramming interface are automatically formalised (see Chapter 8).

The diagramming interface in AKT uses the grammar to form a precise formal statement without the user needing to understand the syntax of that formal statement. Diagrams developed on the basis of this syntax display the following features:

- every node in the diagram represents an object, a process, an action or an attribute of an object or process;
- every node in the diagram is fully labelled;
- information is attached primarily to links rather than nodes;
- the meaning of every link is explicitly stated;
- a pair of linked nodes represents information that corresponds to a unitary statement.

4.4.2 NODES

4.4.2.a Classification of nodes

Four types of node are identified:

Object nodes represent things, or more commonly, groups of things. An object is something that occurs physically e.g. an oak tree, or in an abstract sense, e.g.climate.

Process nodes represent things that happen e.g. seed germination or soil erosion. Depending on the time scale involved, a process might be called an event (it is clearer to refer to death of an organism as an event rather than a process) but an event is simply a special type of process that occurs instantaneously. From the point of view of the semantics of a diagram, they are equivalent.

Action nodes refer to management actions. An action is a type of process, and actions could be represented as processes, however, an action is distinct in that there is a human directly responsible for causing the process to occur.

Attribute nodes refer to a particular attribute of a process or an object. An attribute is something relating to an object or process that might be measured e.g. height of oak tree, rate of germination, colour of flower. Experience shows that attribute nodes are more frequently used than object, action or process nodes.

4.4.2.b Labelling of nodes

Node names that contain implicit meaning are ambiguous. For example, nodes in influence diagrams might be given the name of objects (e.g. pests) or processes (e.g. grain set), while the links mean ‘influence(s)’. However, ‘pest influence grain set’ is incomplete information: all that it explicitly states is that some attribute of pests influences some attribute of grain set.

Where this attribute is known, it has to be stated rather than implied, e.g. ‘pest population size’ ‘influences’ ‘timing of grain set’ if knowledge is to be explicitly stated. Complete information about the meaning of each node is critical for explicit representation of knowledge in the diagram.

4.4.3 LINKS

As in the statement interface, there are two types of link, the reserved term links (in this case only the reserved term links *causes1way* and *causes2way* can be represented diagrammatically) and the user defined links.

4.4.3.a Attaching information to links

It is a normal convention in creating node and link diagrams to attach the information to nodes, so that links are rarely explicitly labelled. This approach may be intuitive but imposes constraints on the diagram. When this approach is used in constructing, for example, causal diagrams the links simply means ‘causes’, while exactly what is caused is specified in the nodes. So, for example, the node ‘increase in leaf area’ might be linked by a ‘causes’ link to ‘increased crop yield’. This severely constrains further additions to the diagram. The fact that pest attack reduces leaf area might need to be included. This would only now be possible by adding a further node to the diagram ‘decreased leaf area’. It is not desirable to have two nodes referring to leaf area, one to increase and one to decrease. Changes are not necessarily only expressed in terms of changes in the quantity of an attribute. A change may also occur in the presence or absence of an object or process. Under this scheme there may be clumsily labelled nodes e.g. ‘occurrence of ovulation’ or ‘disappearance of frost’.

These problems are overcome by attaching the information about changes occurring to the link rather than the nodes. So ‘increase in leaf area’ ‘causes’ ‘increase in crop yield’ becomes 'leaf area' 'increase in causes increase in' 'crop yield' (see Figure 4.1). Over a large diagram the discipline of attaching the majority of information to the link simplifies diagram construction.

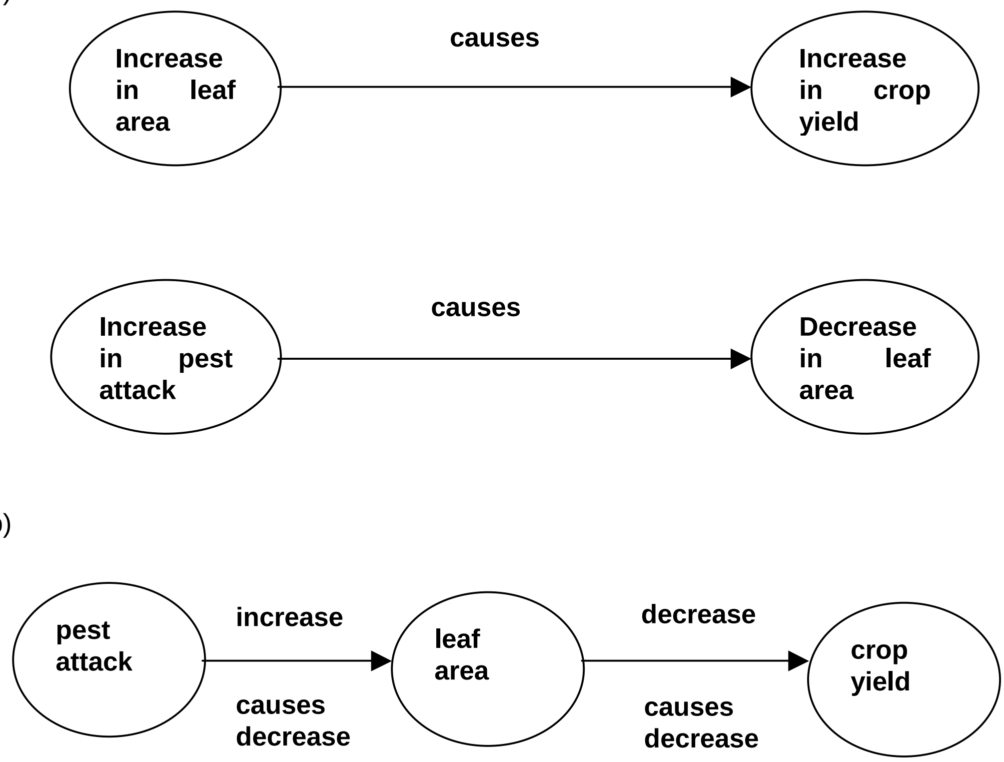


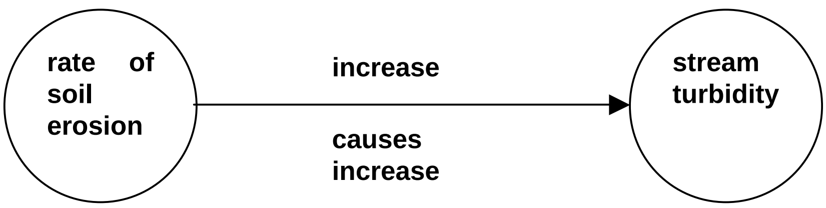
Figure 4.1 *Two ways of expressing the same thing; a) shows the information attached to the nodes whilst b) shows the bulk of the information attached to the links and the advantages in reducing the number of nodes that ensues.*

4.4.3.b Stating the meaning of links

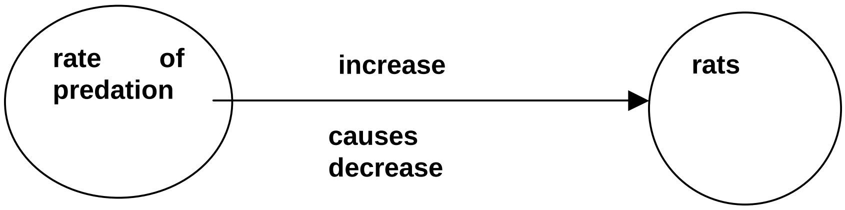
If a diagram is to represent an explicit statement of knowledge, links can only remain unlabelled if they all have exactly the same meaning and this meaning is clearly defined. This makes the mixing of different types of link in a single diagram impossible and is extremely restrictive where the links are considered to be the primary source of information. The explicit statement of the meaning of each link facilitates the flexible development of a representative diagram.

4.4.3.c Linguistic correspondence of linked pairs of nodes.

If two nodes connected by a link represent useful knowledge, there will be a correspondence to a meaningful English sentence. This sentence will, in fact, comply with the definition of a unitary statement (see Part 4.1.2). Where this correspondence does not occur, either one or both nodes or the link must be inappropriately labelled.



can be read as ‘An increase in the rate of soil erosion’ ‘causes’ ‘an increase in stream turbidity’ which makes sense in English.



would read as ‘An increase in the rate of predation’ ‘causes’ ‘a decrease in rats’. This is understandable but incomplete. To formalize this statement one needs to be clear what it is about the rats that decreases, - in this case it is the *number* of rats or their *population size* which decreases, not the rats themselves. Thus this should be explicitly stated by re-labelling ‘Rats’ as ‘Rat population size’ or ‘Rat number’.

The node ‘rate of predation’ is also incompletely labelled and clearly refers to the rate of predation on the rat population. It can be argued that this can be assumed from the context of the diagram, but, where a diagram is created as a means of generating a coherent knowledge base, this cannot be assumed. Predation of other species may occur elsewhere in the knowledge base which can probably not be assumed as being of the same nature as, and therefore equivalent to, the predation of rats. Therefore the first node should properly read ‘Rate of predation on rat population’.

4.4.3.d Labelling of links

It will often be possible to express the meaning of a particular link in different ways. However, using different terms for the same concept reduces the comparability of the resulting diagrams, particularly given that differing terms for the same concept will rarely be exactly equivalent.

Fungal wilt causes yield loss

is similar to:

Fungal wilt results in yield loss

and:

Fungal wilt leads to yield loss

and not very different from:

Fungal wilt is followed by yield loss

which might be restated as:

Fungal wilt happens before yield loss

The above example illustrates the importance of ensuring a consistent use of terms both within a diagram and between diagrams in order to attain unambiguous comparability of links.

4.4.4 SUBSETS OF DIAGRAMS

Experience in the creation of diagrams as a means of producing coherent knowledge bases suggests that diagrams containing more than 40 nodes tend to be increasingly difficult to interpret. This is in part the result of the balance that must be achieved between the size and therefore legibility of nodes and text associated with links, and standard paper or computer screen size. However, even where large boards or paper are used, diagrams rapidly become unsustainably complex. It is the clear representation of links rather than nodes that presents difficulties. Typically a node will be linked to between two and five other nodes. With an increasing number of nodes it becomes increasingly difficult to place nodes near to all the other nodes to which they are linked: links have to travel further across the diagram, crossing increasing numbers of nodes and links until the diagram becomes impenetrable. The placement of nodes, the technique used for representing the meaning of a link and the arrow type used to link nodes all have an impact on the number of nodes and links that can be successfully represented, nevertheless, fundamental limitations will still be reached.

In order to overcome some of these problems, sub-diagrams can be built up from the main diagram. It is possible to make a sub-diagram of the pathway between any two nodes on the diagram. It is also possible to build up a diagram by selecting any number of nodes greater than one, for which the system will then produce a sub-diagram showing all the connections.

4.5 CREATING KNOWLEDGE BASES THROUGH THE COMBINATION OF EXISTING KNOWLEDGE BASES

In parts 4.1 – 4.4 the creation of knowledge bases from scratch has been described. However, it is also possible, and may frequently be desirable, to create a knowledge base through the combination of two or more existing knowledge bases, for example, knowledge in the same domain but from different sets of sources.
In principle the combination of two knowledge bases is a straightforward task. The process of merging knowledge bases is essentially one of ensuring consistency of terminology.

Although effective software support for merging knowledge bases is available, the process is demanding for two reasons. In the first place, two knowledge bases in the same domain are unlikely to use precisely the same terminology, especially where created by different knowledge base developers. Although many equivalent terms (for example, alternative names for the same species) can be identified without difficulty, equivalence for other terms is much more problematic and, even where definitions of terms are provided, can only be resolved through reference to the original knowledge base developer. As a result, it is doubtful that consistent use of terminology across knowledge bases is practically possible unless strict protocols for the use of terminology in the creation of the knowledge bases in the first instance are developed and applied.

A further problem is encountered where the content of the knowledge bases to be merged has been previously formally represented. The process of formal representation involves many decisions about the most appropriate approach to representation. When this process has been undertaken by different individuals for different knowledge bases these decisions will not have been consistently applied to both sets of knowledge. The implications of inconsistency in formal representation for automated reasoning are likely to be serious.

Thus, although it is possible to create a large knowledge base by merging two or more smaller knowledge bases, it is much simpler and much more feasible to begin with a large knowledge base and then split it into smaller ones.

Key points of Chapter Four

- Unitary statements are the basic units of statements in a knowledge base.
- Contextual information can be stored with knowledge statements.
- The definitive clause grammar has been tailor-made for representing knowledge in statements that can be processed by the computer.
- Six elements form the basis for representing knowledge: object, process, action, attribute, link and value.
- Formal statements can be one of the following: attribute-value statement, negative attribute-value statement, causal statement, link statement, comparison statement.

- Determining the elements (formal terms specification), the structure of the statement and statement type are important steps in formalising a knowledge statement.
- Knowledge statements can be entered using the diagram interface through nodes and links.
- Knowledge bases can be split and combined as and when necessary.