

## The Lean Cuisine+ Notation Revised

Chris Scogings

*I.I.M.S., Massey University Albany Campus, Auckland, N.Z.*

*C.Scogings@massey.ac.nz*

### Abstract

The Lean Cuisine+ notation was developed by Chris Phillips (1995) as an executable semi-formal graphical notation for describing the underlying behaviour of event-based direct manipulation interfaces. Lean Cuisine+ builds on the original Lean Cuisine notation introduced in Apperley & Spence (1989). During the construction of a CASE tool for the notation, as well as further research into the use of the Lean Cuisine+ notation, various changes have been made and the revised notation is presented here. The format and much of the content of this document follows that of Appendix C of Phillips (1993) in order to easily distinguish between the earlier and later versions of the notation.

### The Object Model

The object model has been removed from the Lean Cuisine+ notation in order to improve the clarity of the diagrams, since boxes around menemes as well as subtypes are no longer required. The distinction between actions, syntactic objects and semantic objects was often difficult to determine, particularly in the automatic generation of Lean Cuisine+ diagrams for existing interfaces. The work on the automatic generation of Lean Cuisine+ diagrams has been published in Phillips & Scogings (1997) and Scogings & Phillips (1998). Since the object structure of the interface has been removed from the Lean Cuisine+ dialogue tree, it needs to be described elsewhere. Further work is required to decide on the most suitable notation for this, but one possibility would be to use UML case diagrams as described in Fowler (1997).

### 1. Basic features and definitions:

#### Dialogue

A *dialogue* is described by a set of selectable primitives, *menemes*, arranged in a tree structure, and a set of *constraints* over these primitives. The total dialogue state at any time is the sum of all the meneme states plus the values of any *state variables*.

The top level or *root dialogue*, when made available through selection of its *dialogue header* (the root node), provides access to the *subdialogue(s)* and/or *state variable values* immediately below it, subject to any constraints applying.

#### Menemes

The *meneme* is the dialogue primitive, and is an individual selectable representation of an object, operation, state, or value. Menemes may be real or virtual.

*Real* menemes represent specific selectable options, and may be terminal or non-terminal. Non-terminal real menemes are headers to further subdialogues. *Virtual* menemes are always non-terminal and are used to partition a dialogue or subdialogue into further constituent syntactic meneme groupings. They are not available for selection. It is possible to depict the selection of a virtual meneme in a task action sequence but this indicates that one or more of the real menemes within the subdialogue may be selected at that point in the task. See *Task Action Sequences* below.

In each dialogue state, a meneme is either *available* (for selection) or *unavailable*, and either *selected* or *not selected*. This gives rise to four possible meneme states. The state may be changed either by direct excitation, or by indirect modification, subject to any constraints applying. Menemes may be *bistable* (default), *select-only*, *deselect-only*, *monostable* or *passive*. The default type may be explicitly overridden by

specifying one of the other types through the use of *meneme designators* (which are summarised at the end of this section).

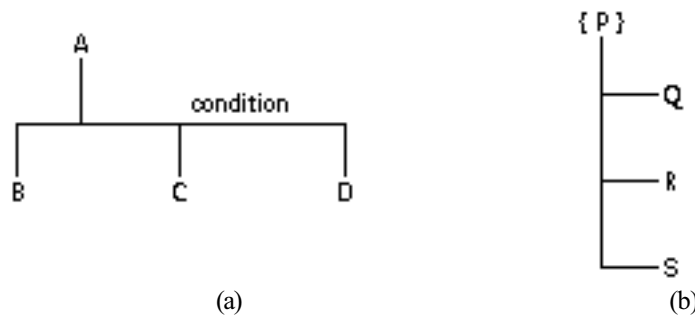
Each meneme can have attributes attaching to it. These can be either *meneme state variables* (MSVs) or *selection preconditions*. MSVs can also exist as parameters to options which may modify them as dialogue progresses. The exact form of presentation of MSVs on a Lean Cuisine+ tree diagram is implementation dependent.

A *selection precondition* is a precondition for the availability (for selection) of a meneme in the Lean Cuisine+ notation, which may involve the state of a meneme in another part of a dialogue, or the existence of some other state or condition, or both. The exact form of presentation is implementation dependent.

**Subdialogues**

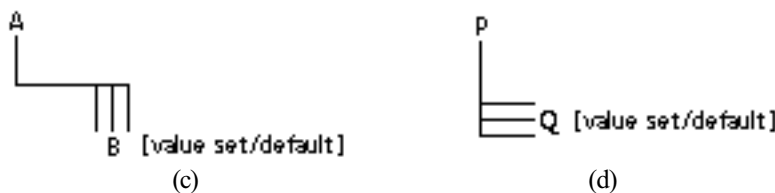
Menemes may be grouped into *subdialogues* to any level of nesting. A subdialogue is a logical grouping of menemes within the body of a Lean Cuisine+ dialogue. The subdialogue corresponding to a non-terminal meneme in a dialogue tree consists of all the menemes with which the *subdialogue header* is connected by downward directed branches. When made available through selection of its header, a subdialogue provides access to the options represented by these menemes, subject to any constraints applying. A subdialogue is available for selection only if its parent dialogue is active.

The grouping of menemes within a subdialogue places constraints on their behaviour. The grouping may be either *mutually exclusive* (1-from-N) or *mutually compatible* (M-from-N). In the latter case, selection may be governed by a condition. In the Lean Cuisine+ notation these structures are represented graphically as follows:



- (a) A mutually compatible meneme group within the subdialogue headed by the real meneme A.
- (b) A mutually exclusive meneme group within the subdialogue headed by the virtual meneme P.

Where a group of options constitute a homogeneous set, they may be represented graphically using a *fork* symbol, as shown below:



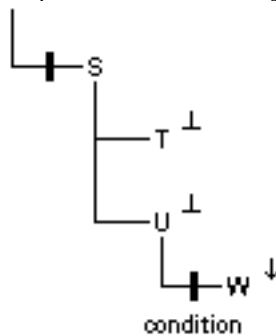
- (c) The tree diagram for a mutually compatible homogeneous group.
- (d) The tree diagram for a mutually exclusive homogeneous group.

*Modal subdialogues*

A subdialogue may be *modal*, in which case, following its selection, options in all other parts of the dialogue become unavailable. A modal subdialogue which involves a subtree is distinguished by a bar across the arc above the meneme which heads it. An example (S) appears in Figure (e).

A modal subdialogue may be *simple*, in which case it is represented by a *monostable* meneme, shown thus ( $\perp$ ). An example (T) appears in Figure (e). Following selection, a meneme of this type reverts to an unselected state on completion of the operation it represents, through system action.

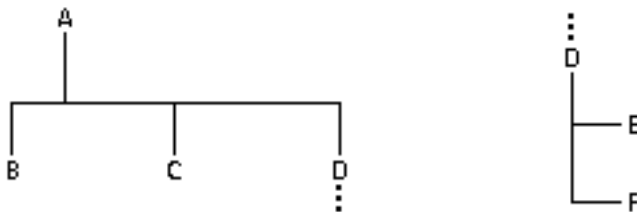
Simple subdialogues are normally terminal nodes. Alternatively they may have explicit modal subdialogues associated with them which are triggered under certain conditions. In such cases the operation represented by the monostable meneme becomes suspended pending user termination of the further modal subdialogue, when it completes in the usual way. An example (U) appears in Figure (e).



- (e) S represents a modal subdialogue, and T a simple modal subdialogue.  
U represents a simple modal subdialogue which under certain conditions triggers a further modal subdialogue W requiring user action.

*Stepwise refinement*

To support *stepwise refinement*, a dialogue or subdialogue tree may be represented visually by an ellipsis to indicate that the detail is developed in a separate diagram. In the case of a subdialogue, the lower level diagram is headed by an ellipsis in order to visually complete the association:

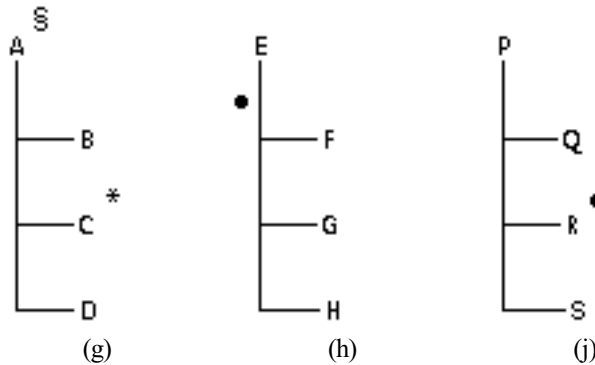


- (f) Meneme D heads a subdialogue which is separately developed.

*Further subdialogue constraints*

A subdialogue header may be tagged to indicate a *required choice* group (§). This places the additional constraint on the relationships between the menemes of that group that a valid selection is always required. An initial *default choice*, which indicates a meneme that is to be initially in the selected state, must be shown (\*) under these circumstances (see Figure (g) below). A *dynamic default* is a default which takes on the value of the last user selection from that group (•). It may have an initial assignment, or it may be

initially unassigned, in which case the first user choice from that group becomes its first value (see Figures (h) and (j) below).



- (g) A is a required choice group, and C is the initial default.  
 (h) An unassigned dynamic default applying to the subgroup FGH.  
 (j) A dynamic default applying to the group QRS, initially assigned to R.

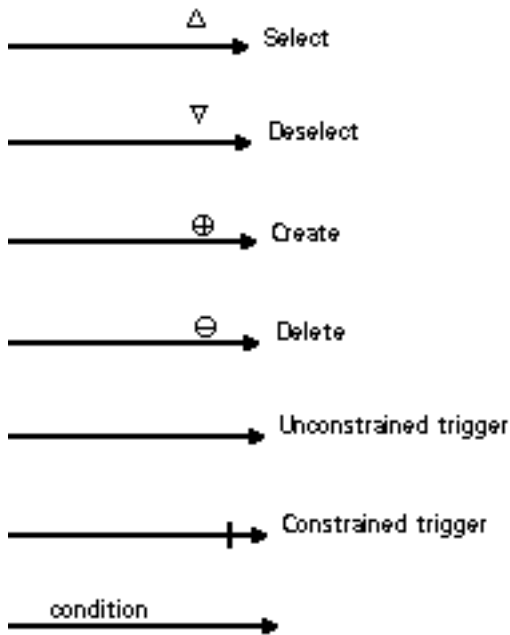
### Meneme designators

The following meneme designators are used in the Lean Cuisine+ notation.

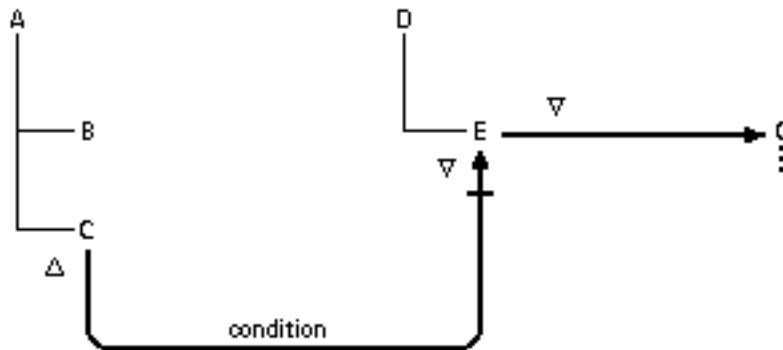
- { } *Virtual meneme*: a non-selectable meneme which is used to partition a dialogue or subdialogue. The meneme name appears between the braces.
- ⊥ *Monostable meneme*: a meneme which, following user selection, reverts to an unselected state on completion of the simple modal subdialogue it represents, through system action.
- ↑ *Select-only meneme*: a meneme of this type cannot be directly deselected by the user, but may be deselected through selection of another mutually exclusive option.
- ↓ *Deselect-only meneme*: a meneme of this type cannot be directly selected by the user, but may become selected as an indirect result of some other action.
- ⊗ *Passive meneme*: a meneme of this type is unavailable for selection or deselection by the user, and can only be selected and/or deselected by the system.
- \* *Default choice*: indicates a meneme that is to be in the selected state at the commencement of a dialogue or subdialogue.
- § *Required choice*: is associated with a subdialogue header and indicates a meneme group from which a valid choice is always required.
- *Dynamic default*: may be associated with a meneme (assigned) or a meneme group (unassigned) and indicates a default choice which takes on the value of the last user selection from that group.

### 2. Selection Triggers

A *selection trigger* is a directed link between menemes such that the selection or deselection of a meneme triggers (possibly conditionally) the selection or deselection of one or more other menemes. Selection triggers represent a system response to another action and are annotated as shown below.



Triggers are either *unconstrained*, in which case further "knock on" effects are possible, or *constrained*, in which case the behaviour of the selected or deselected meneme is modified, inhibiting further links, and preventing side effects such as cycles. An example appears in Figure (k).



- (k) Selection ( $\Delta$ ) of C causes deselection ( $\nabla$ ) of E provided the condition is true. The selection of G is not triggered because the first trigger is constrained.

### 3. Task Action Sequences

A *task action sequence* describes a higher level user task in terms of primitive actions (and associated events), capturing any temporal relationships between them. The task action sequence is overlayed on the dialogue tree diagram and is orthogonal to the constraint-based behavioural interface model.

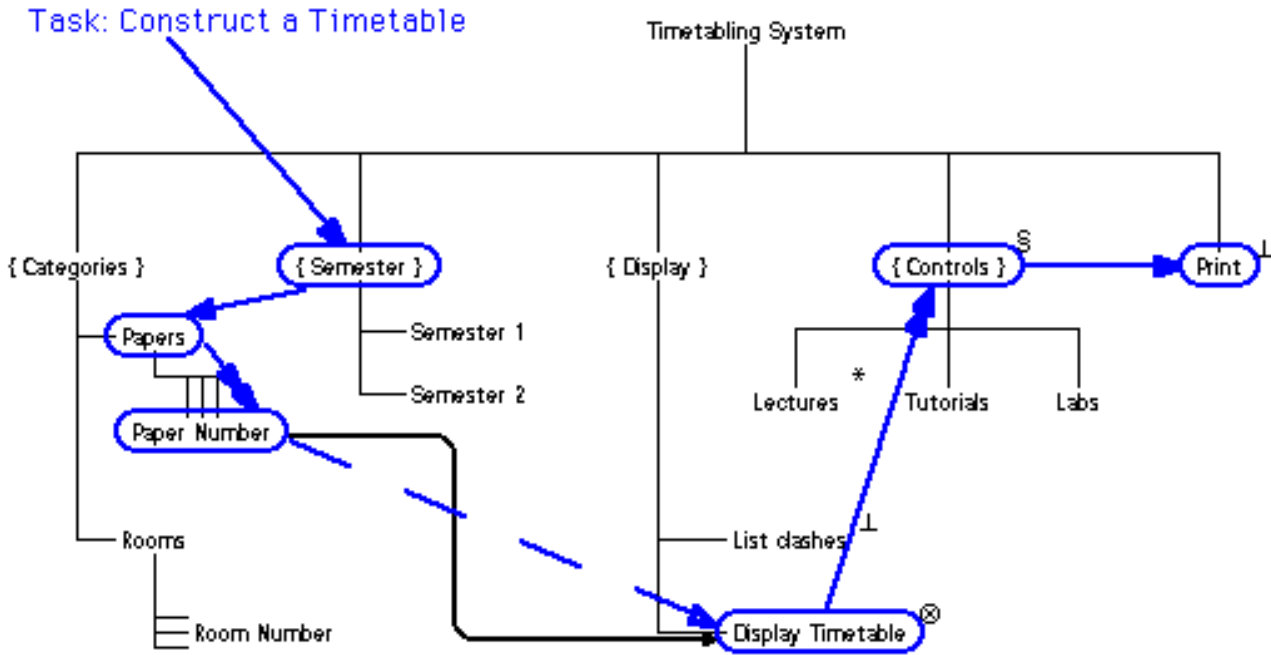


Figure 1. The task action sequence for the task "Construct a Timetable". The selection of the virtual meneme "Controls" actually indicates the selection of at least one of the menemes in the subdialogue. The double-headed arrow indicates that any combination of the menemes in the subdialogue may be selected. The dashed arrow indicates the system selection of "Display Timetable" in response to the selection trigger.

The notation distinguishes user initiated and system initiated events by means of the arrows connecting selections. A solid arrow between two selections denotes that the new selection is the result of a user action, whereas a dashed arrow indicates that the new selection is a system response triggered by some other action. Selection and deselection actions are indicated as defined above for selection triggers.

A virtual meneme can be shown as being selected in a task action sequence. This implies the selection of any allowable combination of menemes in the subdialogue below it. The double headed arrow indicates that an action may be repeated. This may be used in conjunction with the selection of a virtual meneme.

The example provided in Figure 1 above, as well as further discussion on the usefulness of the new Lean Cuisine+ task sequence notation, have been published in Phillips & Scogings (2000).

**4. Execution of a Lean Cuisine+ specification**

A Lean Cuisine+ dialogue diagram can be "marked" to show its state at a given point in the dialogue in terms of meneme states. Menemes can be shown as being either selected or unselected, and either available for selection or unavailable. This provides for the "execution" of a Lean Cuisine+ specification - that is of simulating dynamically the behaviour of the dialogue as selections are made. This process takes cognisance of all specified constraints, such as selection triggers.

The exact form of presentation of the "execution" process within the context of a Lean Cuisine+ tree diagram is implementation dependent.

## References

Apperley, M.D. & Spence, R. (1989): Lean Cuisine: A low-fat notation for menus, *Interacting with Computers*, vol 1, no 1.

Fowler M. & Scott K. (1997): *UML Distilled*, Addison-Wesley, Reading, Massachusetts

Phillips C.H.E. (1993): The Development of an Executable Graphical Notation for Describing Direct Manipulation Interfaces, *PhD Thesis*, Massey University.

Phillips C.H.E. (1995): Lean Cuisine+: An Executable Graphical Notation for Describing Direct Manipulation Interfaces, *Interacting with Computers*, vol 7, no 1.

Phillips C.H.E. & Scogings C.J. (1997): Modelling the mock-up: towards the automatic specification of the behaviour of early prototypes, *Proc. INTERACT'97*, Sydney.

Phillips C.H.E. & Scogings C.J. (2000): Task and Dialogue modelling: Bridging the divide with Lean Cuisine+, *Australian Computer Science Communications*, vol 22, no 5.

Scogings C.J. & Phillips C.H.E. (1998): Beyond the interface: Modelling the Interaction in a Visual Development Environment, *Proc. HCI'98*, Sheffield.

