NCClusterCard

By: Randy Trigg (Trigg.pa@Xerox.COM)

Stored: {qv}<notecards>1.3L>Library>NCClusterCard, .lcom, .ted
Last updated: March 17, 1988 by Peggy Irish.
See also NCCaseCluster and NCScreen packages.

INTRODUCTION

Loading this package on top of NoteCards causes the creation of the Cluster card type. Unlike other card types, creating a Cluster causes an interlinked network of cards to be created. One of these cards is called the head cluster card. In some sense all control rests with this head card - the other members of the structure aren't "aware" that they are participating in a larger network. Operations that need to be performed to the cluster as a whole are directed from the head card. In addition to the head card there are one or more cluster children cards which can be of any type and are linked together in some manner at cluster creation time.

Currently there is no way for the user to create an instance of the generic Cluster type since it doesn't appear in the global NewCards menu. In that sense, Cluster serves as a "mixin" for specialized cluster types. For an example of the use of Cluster cards in legal case analysis, see the CaseCluster library package.

MANIPULATING CLUSTERS FROM THE USER INTERFACE

Because the head cluster card "knows" who its children are, certain operations can be performed on the entire group at once. This is always done from the head cluster card - individual child cards of the cluster can be manipulated independently in the usual ways.

The left button title bar menu of the head cluster card has an entry called LayoutClusterChildren. This allows the user to bring up the child cards of the cluster and position them on the screen according to various layout regimes. The default layout regime, Cascade, causes the cards to be stacked under the head cluster card like a deck of cards. The pull-across menu for LayoutClusterChildren also contains items for layout according to Square and Surround strategies. Layout in Square causes the cards to be positioned in a non-overlapping manner on the screen, if

possible. Surround causes the cards to be laid out around three sides of the head card. (For details on all three of these layout regimes, see the documentation for the NCScreen library package.)

If the head cluster card is closed, then any open children will also be closed. If the head cluster card is brought up (by following a link to it), then its children are also brought up and arranged in a cascade.

SPECIALIZING CLUSTER CARDS

The predominant use of Cluster cards is for other card types to specialize. (See the CaseCluster library package for an example of how this has been done in the legal case analysis domain.) In order to create tailored application-specific clusters, one needs to create a new card type which inherits from Cluster. (The Cluster card type itself inherits from Text.) Certain aspects of the cluster structure have been parameterized. In particular, the MakeFn of the Cluster card type accepts certain structural directives packaged in the form of its fourth argument.

(CLUSTER.MakeFn <Card> <Title> <NoDisplayFlg> <ClusterInfo>)

The first three arguments are the normal arguments to any card type's MakeFn. The <ClusterInfo> argument should be in the form of a list of two elements or if you prefer, an instance of the CLUSTER.ClusterInfo record. The first field is called ChildInfos and the second, FileBoxes. The ChildInfos field contains a list of specifications for the child cards while the FileBoxes field gives fileboxes in which to file new child cards as they are created. The length of FileBoxes should be no longer than the length of ChildInfos. (NIL entries cause the corresponding child not to be filed at creation time.)

ChildInfos is a list of three-element lists or if you prefer, instances of the CLUSTER.ChildInfo record. Each ChildInfo consists of the three fields CardType, LinkType, and TitleString. When an instance of the cluster is created, one child is created for each ChildInfo record in the list. The child has type CardType and is linked from the head cluster card with a link of type LinkType. (NIL LinkType means no link from the head cluster card is created.) Its title is constructed by concatenating the title of the head cluster card with the string TitleString. (If TitleString is NIL, then the print name of CardType is used.)

Thus, following execution of CLUSTER.MakeFn, several cards have been created: a head cluster card of type Cluster and several child cards of possibly different types. These are linked from the head card using possibly different link types. In addition, the head cluster card explicitly stores the UIDs of each of its child cards. (Thus,

deleting a link from the head card does NOT remove that child from the cluster.)
The other methods of the Cluster card type work in the standard way (as Text cards) with the following exceptions. The EditFn of the Cluster card causes the child cards to be brought up and arranged in a cascade under the head card. The QuitFn closes any open child cards in addition to the head card. The PutFn and GetFn write and read the set of child UIDs before writing and reading the text substance of the head card. Note also that the Cluster card type does not appear on the card types menu since there is as of yet no interactive interface for users to specify the structure of the cluster.