

File created: 6-Oct-88 16:38:13 {DSK}<LISPPFILES>NOTECARDS>NCTOULMINCARD.;6

changes to: (FNS NCToulmin.WhenDeletedFn NCToulmin.MakeFn NCToulmin.AddToulminCard  
NCToulmin.RelayoutToulminCard NCToulmin.EditFn)  
(VARS NCTOULMINCARDCOMS)

previous date: 4-Oct-88 10:12:22 {DSK}<LISPPFILES>NOTECARDS>NCTOULMINCARD.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1988 by Xerox Corporation. All rights reserved.

(RPAQQ NCTOULMINCARDCOMS  
[

;;; This card type is used to represent the Toulmin argument structure. Upon creation of a Toulmin card, the user is asked to specify the cards which  
;;; make up the components (Datum, Claim, Warrant, Backing, and Rebuttal) of the argument, either by selecting existing cards, or creating new ones.  
;;; Selecting "Cancel" at any time aborts the creation of the Toulmin card.

;;; The user specifies the placement of the links to these cards and optional labels in the Toulmin card through the NCToulmin.NodeInfos and  
;;; NCToulmin.LabelInfos globalvars. Edges among the links and labels are specified through NCToulmin.EdgeInfos.

```
(DECLARE%: DONTCOPY (PROPS (NCTOULMINCARD MAKEFILE-ENVIRONMENT)
                           (NCTOULMINCARD FILETYPE)))
[DECLARE%: DONTEVAL@LOAD FIRST (P (NC.LoadFileFromDirectories 'NCGRAPHCARD)
(GLOBALVARS NCToulmin.DefaultHeight NCToulmin.DefaultWidth NCToulmin.DefaultFileBoxName
            NCToulmin.NodeInfos NCToulmin.LabelInfos NCToulmin.EdgeInfos NCToulmin.LargeFont
            NCToulmin.FileBoxNames NCToulmin.HashArray NC.ToulminCardIcon)
(INITVARS (NCToulmin.DefaultHeight 330)
          (NCToulmin.DefaultWidth 500))
```

;;; The NCToulmin.NodeInfo record definition is: (RECORD NCToulmin.NodeInfo (Name CardType LinkType Title Position)). Name is the name of the  
;;; component node, CardType is the type of the card to be used for the component, and LinkType is the type of link to be made between the Toulmin  
;;; card and the component card. Title is the title to be used for a new card created to represent the component. Position is the position in the Toulmin  
;;; card at which the link to the component will be placed.

```
(INITVARS [NCToulmin.NodeInfos '((Datum Text Datum NIL (75 . 275))
                                (Claim Text Claim NIL (415 . 275))
                                (Warrant Text Warrant NIL (245 . 165))
                                (Backing Text Backing NIL (245 . 35))
                                (Rebuttal Text Rebuttal NIL (415 . 165))
                                (NCToulmin.HashArray (HASHARRAY 100)))
```

;;; The NCToulmin.LabelInfos record definition is: (RECORD (NCToulmin.LabelInfo (LabelName LabelString LabelPosition LabelFont LabelBorder  
;;; LabelShade)). LabelName is the name of the label. LabelString is the string to appear in the Toulmin card. LabelPosition is the position of the label.  
;;; LabelFont is the font in which the label should appear. LabelBorder should be T if a border (box) is desired around the label, otherwise NIL.  
;;; LabelShade is the background shade the label should have.

```
[INITVARS (NCToulmin.LargeFont (FONTCREATE 'HELVETICA 14 'MRR))
          (NCToulmin.LabelInfos `((Dot "." (245 . 275)
                                ,NCToulmin.LargeFont)
                                (So "[So]" (300 . 275)
                                ,NCToulmin.LargeFont)
                                (Since "[Since]" (245 . 220)
                                ,NCToulmin.LargeFont)
                                (OnAccountOf "[On Account Of]" (245 . 100)
                                ,NCToulmin.LargeFont)
                                (Unless "[Unless]" (415 . 220)
                                ,NCToulmin.LargeFont]
```

;;; The NCToulmin.EdgeInfo record definition is: (RECORD NCToulmin.EdgeInfo (EdgeName SourceNodeName DestinationNodeName)). EdgeName  
;;; is the name of the edge. SourceNodeName is the name of the node (link or label) at the source end of the edge. DestinationNodeName is the name  
;;; of the node at the destination end of the edge.

```
[INITVARS (NCToulmin.EdgeInfos '((Edge1 Datum Dot)
                                (Edge2 Dot So)
                                (Edge3 So Claim)
                                (Edge4 Dot Since)
                                (Edge5 Since Warrant)
                                (Edge6 Warrant OnAccountOf)
                                (Edge7 OnAccountOf Backing)
                                (Edge8 Claim Unless)
                                (Edge9 Unless Rebuttal]
[INITVARS (NCToulmin.DefaultArgumentBoxName 'ArgumentBox)
          (NCToulmin.FileBoxNames '(DatumBox ClaimBox WarrantBox BackingBox RebuttalBox]
(RECORDS NCToulmin.NodeInfo NCToulmin.LabelInfo NCToulmin.EdgeInfo)
```

```
(BITMAPS NC.ToulminCardIcon)
(FNS NCToulmin.MakeFn NCToulmin.EditFn NCToulmin.WhenDeletedFn)
(FNS NCToulmin.AddToulminCard NCToulmin.RelayoutToulminCard)
(DECLARE%: DONTEVAL@LOAD (P (NCToulmin.AddToulminCard)
                             (PUTPROP 'Toulmin 'WhenDeletedFn (FUNCTION NCToulmin.WhenDeletedFn))
```

;;; This card type is used to represent the Toulmin argument structure. Upon creation of a Toulmin card, the user is asked to specify the cards which  
 ;;; make up the components (Datum, Claim, Warrant, Backing, and Rebuttal) of the argument, either by selecting existing cards, or creating new ones.  
 ;;; Selecting "Cancel" at any time aborts the creation of the Toulmin card.

;;; The user specifies the placement of the links to these cards and optional labels in the Toulmin card through the NCToulmin.NodeInfos and  
 ;;; NCToulmin.LabelInfos globalvars. Edges among the links and labels are specified through NCToulmin.EdgeInfos.

```
(DECLARE%: DONTCOPY

(PUTPROPS NCTOULMINCARD MAKEFILE-ENVIRONMENT (:PACKAGE "INTERLISP" :READTABLE "INTERLISP"))

(PUTPROPS NCTOULMINCARD FILETYPE :BCOMPL)
)

(DECLARE%: DONTEVAL@LOAD FIRST

(NC.LoadFileFromDirectories 'NCGRAPHCARD)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NCToulmin.DefaultHeight NCToulmin.DefaultWidth NCToulmin.DefaultFileBoxName NCToulmin.NodeInfos
             NCToulmin.LabelInfos NCToulmin.EdgeInfos NCToulmin.LargeFont NCToulmin.FileBoxNames NCToulmin.HashArray
             NC.ToulminCardIcon)
)

(RPAQ? NCToulmin.DefaultHeight 330)

(RPAQ? NCToulmin.DefaultWidth 500)
```

;;; The NCToulmin.NodeInfo record definition is: (RECORD NCToulmin.NodeInfo (Name CardType LinkType Title Position)). Name is the name of the  
 ;;; component node, CardType is the type of the card to be used for the component, and LinkType is the type of link to be made between the Toulmin  
 ;;; card and the component card. Title is the title to be used for a new card created to represent the component. Position is the position in the Toulmin  
 ;;; card at which the link to the component will be placed.

```
(RPAQ? NCToulmin.NodeInfos
'((Datum Text Datum NIL (75 . 275))
  (Claim Text Claim NIL (415 . 275))
  (Warrant Text Warrant NIL (245 . 165))
  (Backing Text Backing NIL (245 . 35))
  (Rebuttal Text Rebuttal NIL (415 . 165))))
```

```
(RPAQ? NCToulmin.HashArray (HASHARRAY 100))
```

;;; The NCToulmin.LabelInfos record definition is: (RECORD (NCToulmin.LabelInfo (LabelName LabelString LabelPosition LabelFont LabelBorder  
 ;;; LabelShade)). LabelName is the name of the label. LabelString is the string to appear in the Toulmin card. LabelPosition is the position of the label.  
 ;;; LabelFont is the font in which the label should appear. LabelBorder should be T if a border (box) is desired around the label, otherwise NIL.  
 ;;; LabelShade is the background shade the label should have.

```
(RPAQ? NCToulmin.LargeFont (FONTCREATE 'HELVETICA 14 'MRR))
```

```
(RPAQ? NCToulmin.LabelInfos
'((Dot "." (245 . 275)
   ,NCToulmin.LargeFont)
 (So "[So]" (300 . 275)
   ,NCToulmin.LargeFont)
 (Since "[Since]" (245 . 220)
   ,NCToulmin.LargeFont)
 (OnAccountOf "[On Account Of]" (245 . 100)
   ,NCToulmin.LargeFont)
 (Unless "[Unless]" (415 . 220)
   ,NCToulmin.LargeFont)))
```

;;; The NCToulmin.EdgeInfo record definition is: (RECORD NCToulmin.EdgeInfo (EdgeName SourceNodeName DestinationNodeName)). EdgeName  
 ;;; is the name of the edge. SourceNodeName is the name of the node (link or label) at the source end of the edge. DestinationNodeName is the name  
 ;;; of the node at the destination end of the edge.

```
(RPAQ? NCToulmin.EdgeInfos
'((Edge1 Datum Dot)
 (Edge2 Dot So)
 (Edge3 So Claim)
 (Edge4 Dot Since)
 (Edge5 Since Warrant)
 (Edge6 Warrant OnAccountOf)
 (Edge7 OnAccountOf Backing)
 (Edge8 Claim Unless)
 (Edge9 Unless Rebuttal)))
```

(RPAQ? NCToulmin.DefaultArgumentBoxName 'ArgumentBox)

(RPAQ? NCToulmin.FileBoxNames ' (DatumBox ClaimBox WarrantBox BackingBox RebuttalBox))

(DECLARE%: EVAL@COMPILE

(RECORD NCToulmin.NodeInfo (Name CardType LinkType Title Position))

(RECORD NCToulmin.LabelInfo (LabelName LabelString LabelPosition LabelFont LabelBorder LabelShade))

(RECORD NCToulmin.EdgeInfo (EdgeName SourceNodeName DestinationNodeName))

)

(RPAQQ NC.ToulminCardIcon )

(DEFINEQ

**(NCToulmin.MakeFn**

[LAMBDA (ToulminCard Title NoDisplayFlg) ; Edited 5-Oct-88 17:16 by pmi

;; Create a Toulmin card. Done by defining proper info as well as list of fileboxes through NCToulmin.ChildInfos, NCToulmin.LabelInfos,  
;; NCToulmin.EdgeInfos and NCToulmin.FileBoxNames.

;; The elements of NCToulmin.NodeInfos are instances of the NCToulmin.ChildInfo record. If the TitleString field of the record is NIL, then use the  
;; card's type.

;; pmi 8/18/88: Now prompts for titles of component cards after constructing the Toulmin card. Also, prompt for fileboxes appears on NoteFile icon  
;; instead of main prompt window.

(**DECLARE** (GLOBALVARS NCToulmin.DefaultArgumentBoxName NCToulmin.NodeInfos NCToulmin.LabelInfos  
NCToulmin.EdgeInfos NCToulmin.HashArray NCToulmin.FileBoxNames NC.SelectingSingleCardMenu))

(**PROG** ((NoteFile (NCP.CardNoteFile ToulminCard))  
ContentsFileBox Window ToulminFileBox FileBoxes ComponentCards ComponentCardsToDelete Region  
ReturnValue)  
(SETQ ContentsFileBox (NCP.ContentsFileBox NoteFile))  
(SETQ Window (NCP.CoerceToInterestedWindow NoteFile))

;; This hash array will be used to store the nodes created in the graph for each of the links to component cards of the Toulmin structure.

(CLRHASH NCToulmin.HashArray)

;; Set up the fileboxes for the component cards, if they don't already exist.

[SETQ FileBoxes (**for** FileBoxName **in** NCToulmin.FileBoxNames  
**collect** (OR (NCP.LookupCardByName FileBoxName NoteFile)  
(LET ((FileBox (NCP.CreateFileBox NoteFile FileBoxName T NIL NIL  
ContentsFileBox))  
(NCP.RegisterCardByName FileBoxName FileBox)  
FileBox])

;; Make the underlying graph card.

(SETQ Window (APPLY\* (NCP.CardTypeFn 'Graph 'MakeFn  
ToulminCard Title NoDisplayFlg))

;; Create the component cards.

(SETQ ComponentCards  
(**for** NodeInfo **in** NCToulmin.NodeInfos **as** Box **in** FileBoxes **until** (EQ ReturnValue 'Cancelled)  
**collect** (LET (ComponentCard Node (Name (**fetch** (NCToulmin.NodeInfo Name) **of** NodeInfo))  
(CardType (**fetch** (NCToulmin.NodeInfo CardType) **of** NodeInfo))  
(LinkType (**fetch** (NCToulmin.NodeInfo LinkType) **of** NodeInfo)))  
[**while** (NULL Node) **do** (SETQ ComponentCard (NCP.SelectCards Window T NIL  
(CONCAT "Select a card for the " Name  
" of this Toulmin structure.")  
T T))  
(**if** (EQ ComponentCard 'DON'T)  
**then** (SETQ ReturnValue 'Cancelled)  
(SETQ ComponentCard NIL)  
(SETQ Node T)  
**else** (**if** (EQ ComponentCard '\*New% Card\*)  
**then** (SETQ ComponentCard (NCP.CreateCard  
CardType NoteFile  
(**fetch** (NCToulmin.NodeInfo  
Title)  
**of** NodeInfo)  
T NIL Box))  
**elseif** (NOT (MEMBER ComponentCard (NCP.FileBoxChildren  
Box)))  
**then** (NCP.ChangeLoc Box 'END)  
(NCP.FileCards ComponentCard Box))  
(**if** LinkType  
**then** (OR (NCP.ValidLinkTypeP LinkType NoteFile)  
(NCP.CreateLinkType LinkType NoteFile))  
(SETQ Node (NCP.LocalGlobalLink LinkType  
ToulminCard ComponentCard  
(**fetch** (NCToulmin.NodeInfo  
Position)  
**of** NodeInfo))  
(**if** ComponentCard  
**then** (PUTHASH Name Node NCToulmin.HashArray)

```
(NCP.CardUserDataProp ComponentCard 'Name Name))
ComponentCard))
```

;; If the user chooses to cancel the Toulmin card creation, give them a chance to delete any component cards just created.

```
(if (EQ ReturnValue 'Cancelled)
  then (SETQ ComponentCardsToDelete (for ComponentCard in ComponentCards
    when (AND (NCP.ValidCardP ComponentCard)
              (NCP.NewCardP ComponentCard)
              (NCP.AskYesOrNo (CONCAT "Delete new "
                                     (NCP.CardUserDataProp
                                      ComponentCard
                                      'Name)
                                     " card: "
                                     (NCP.CardTitle ComponentCard
                                      )
                                     )
                                     "? ")
              NIL "No" T Window))
    collect ComponentCard))
  (CLOSEW Window)
  (AND ComponentCardsToDelete (NCP.DeleteCards ComponentCardsToDelete))
  (RETURN NIL))
```

;; Fill in the labels.

```
(for LabelInfo in NCToulmin.LabelInfos do (PUTHASH (fetch (NCToulmin.LabelInfo LabelName) of LabelInfo)
  (NC.InsertLabelInGraph ToulminCard (fetch (
    NCToulmin.LabelInfo
    LabelString)
    of LabelInfo)
  (fetch (NCToulmin.LabelInfo LabelPosition)
    of LabelInfo)
  (fetch (NCToulmin.LabelInfo LabelFont)
    of LabelInfo)
  (fetch (NCToulmin.LabelInfo LabelBorder)
    of LabelInfo)
  (fetch (NCToulmin.LabelInfo LabelShade)
    of LabelInfo)
  T)
  NCToulmin.HashArray))
```

;; Add in the specified edges, using the hash array to find the graph nodes associated with each of the source and destination names in  
;; NCToulmin.EdgeInfos.

```
(for EdgeInfo in NCToulmin.EdgeInfos bind SourceNode DestinationNode
  do (AND (SETQ SourceNode (GETHASH (fetch (NCToulmin.EdgeInfo SourceNodeName) of EdgeInfo)
    NCToulmin.HashArray))
          (SETQ DestinationNode (GETHASH (fetch (NCToulmin.EdgeInfo DestinationNodeName) of EdgeInfo)
    NCToulmin.HashArray))
          (NC.InsertEdgeInGraph ToulminCard SourceNode DestinationNode)))
```

;; Give the user a chance to change the title of each component card.

```
(for ComponentCard in ComponentCards as NodeInfo in NCToulmin.NodeInfos bind OldTitle NewTitle
  NewUntitledFlg
  do (SETQ NewUntitledFlg)
      (SETQ OldTitle (NC.RetrieveTitle ComponentCard))
      (SETQ NewTitle (NCP.AskUser (CONCAT "Enter the title for this " (fetch (NCToulmin.NodeInfo Name)
        of NodeInfo)
        " card"
        (CHARACTER 13))
        "--> "
        (if (AND (STREQUAL OldTitle "Untitled")
                (NCP.NewCardP ComponentCard))
          then (SETQ NewUntitledFlg T)
              NIL
          else OldTitle)
        NIL Window))
      (if (OR (STREQUAL OldTitle NewTitle)
              (AND NewUntitledFlg (NULL NewTitle)))
        else (NCP.CardTitle ComponentCard NewTitle)))
```

;; Title the new Toulmin card

```
(if Title
  then (NCP.CardTitle ToulminCard Title)
  else ;; Ask for a title if it has not already been specified.
      (NCP.PrintMsg Window T "Toulmin card: ")
      (NC.AssignTitle ToulminCard T NIL Window))
```

;; File this new Toulmin card in the proper filebox.

```
(SETQ ToulminFileBox (OR (NCP.LookupCardByName NCToulmin.DefaultArgumentBoxName NoteFile)
  (LET ((FileBox (NCP.CreateFileBox NoteFile NCToulmin.DefaultArgumentBoxName T
    NIL NIL ContentsFileBox)))
    (NCP.RegisterCardByName NCToulmin.DefaultArgumentBoxName FileBox)
    FileBox)))
(NCP.FileCards ToulminCard ToulminFileBox)
```

;; Fix up a few window properties

```
(WINDOWADDPROP Window 'SHRINKFN (FUNCTION NC.GraphCardShrinkFn))
(WINDOWADDPROP Window 'REPAINTFN (FUNCTION NCToulmin.RelayoutToulminCard))
```

```
T)
(RETURN (OR Window ToulminCard])
```

**(NCToulmin.EditFn**

```
[LAMBDA (Card Substance Region/Position) ; Edited 4-Oct-88 17:33 by pmi
;; Given a Toulmin Substance, open a Toulmin card window and set it up to be a NoteCard with ID
(LET ((GraphNode (fetch (GRAPH GRAPHNODES) of Substance))
Window)
;; Bring up card and mess with its window.
(SETQ Window (APPLY* (NCP.CardTypeFn 'Graph 'EditFn
Card Substance Region/Position))
(NC.GraphLinkIconUpdateCheck Card Window Substance T)
;; Fix up a few window properties
(WINDOWADDPROP Window 'SHRINKFN (FUNCTION NC.GraphCardShrinkFn))
(WINDOWADDPROP Window 'REPAINTFN (FUNCTION NCToulmin.RelayoutToulminCard)
T)
Window])
```

**(NCToulmin.WhenDeletedFn**

```
[LAMBDA (ArgumentCard) ; Edited 6-Oct-88 16:27 by pmi
;; Delete each argument element of this argument card if all of the following are true:
;; 1- It is not a part of any other argument,
;; 2- It is not filed in any fileboxes other than those containing argument elements,
;; 3- The user confirms that it should be deleted.
(DECLARE (GLOBALVARS NCToulmin.FileBoxNames))
(LET ((NoteFile (NCP.CardNoteFile ArgumentCard))
FileBoxes ArgumentElements CardsToDelete)
(SETQ FileBoxes (for FileBoxName in NCToulmin.FileBoxNames bind FileBox when (SETQ FileBox
(NCP.LookupCardByName
FileBoxName NoteFile
))
collect FileBox))
[SETQ ArgumentElements (NCP.CardNeighbors ArgumentCard '(Datum Claim Backing Warrant Rebuttal]
(SETQ CardsToDelete (for ArgumentElement in ArgumentElements bind
when (AND [NULL (REMOVE ArgumentCard (NCP.CardNeighbors ArgumentElement
'(_Datum
_Claim
_Backing
_Warrant
_Rebuttal]
(NULL (LDIFFERENCE (NCP.CardParents ArgumentElement)
FileBoxes)))
collect ArgumentElement))
(for Card in CardsToDelete when (NCP.AskYesOrNo (CONCAT "Okay to delete argument element '" (
NCP.CardTitle
Card)
"'? ")
NIL
'No T (NCP.CardWindow ArgumentCard))
do (NCP.DeleteCards Card])
)
```

(DEFINEQ

**(NCToulmin.AddToulminCard**

```
[LAMBDA NIL ; Edited 5-Oct-88 14:02 by pmi
(NCP.CreateCardType 'Toulmin 'Graph `[(MakeFn (FUNCTION NCToulmin.MakeFn]
`([LinkDisplayMode (NCP.CoerceToLinkDisplayMode '(T T FLOAT]
(DisplayedInMenuFlg ,T)
(DefaultHeight ,NCToulmin.DefaultHeight)
(DefaultWidth ,NCToulmin.DefaultWidth)
(LinkIconAttachedBitMap ,NC.ToulminCardIcon)
(MiddleButtonMenuItems NIL])
```

**(NCToulmin.RelayoutToulminCard**

```
[LAMBDA (Window) ; Edited 4-Oct-88 17:31 by pmi
(LET ((Graph (WINDOWPROP Window 'GRAPH))
(GraphCard (NCP.CardFromWindow Window)))
(NC.GraphLinkIconUpdateCheck GraphCard Window Graph T])
)
```

(DECLARE%: DONTEVAL@LOAD

(NCToulmin.AddToulminCard)

(PUTPROP 'Toulmin 'WhenDeletedFn (FUNCTION NCToulmin.WhenDeletedFn))

)

{MEDLEY}<notecards>library>nctoulmincard.;1

Page 6

(PUTPROPS **NCTOULMINCARD COPYRIGHT** ("Xerox Corporation" 1988))

---

**FUNCTION INDEX**

NCToulmin.AddToulminCard .....5    NCToulmin.MakeFn .....3    NCToulmin.WhenDeletedFn .....5  
NCToulmin.EditFn .....5    NCToulmin.RelayoutToulminCard ....5

---

**VARIABLE INDEX**

NC.ToulminCardIcon .....3    NCToulmin.EdgeInfos .....2    NCToulmin.LargeFont .....2  
NCToulmin.DefaultArgumentBoxName .3    NCToulmin.FileBoxNames .....3    NCToulmin.NodeInfos .....2  
NCToulmin.DefaultHeight .....2    NCToulmin.HashArray .....2    NCTOULMINCARDOMS .....1  
NCToulmin.DefaultWidth .....2    NCToulmin.LabelInfos .....2

---

**RECORD INDEX**

NCToulmin.EdgeInfo .....3    NCToulmin.LabelInfo .....3    NCToulmin.NodeInfo .....3

---

**PROPERTY INDEX**

NCTOULMINCARD .....2

---