DOCTOR OF PHILOSOPHY

Use of Human Reliability Analysis to evaluate surgical technique for rectal cancer

Wilson, Peter John

Award date:
2012

DOCTOR OF PHILOSOPHY

Use of Human Reliability Analysis to evaluate surgical technique for rectal cancer

Peter John Wilson

2012

Appendix 6
Applescript Code for Extraction of Data from Text File

```
-- FindErrors.applescript
-- FindErrors

--  Created by Peter Wilson on 15/05/2006.
--  Copyright 2006 __MyCompanyName__. All rights reserved.

on awake from nib theObject
    global MyFol, AllFiles, CountFile, TaskLimits, ErrorList, ToEx, MyTog, Dest,
OpType, OpTypes, TaskList, TaskName, SubTaskName, StoreText
    set ErrorList to {"EEM", "Failure", "Error", "Consequence", "Recovery",
"Preparation", "Techniques", "Other"}
    set ToEx to {"0  0  0  0  0  0  0  0  0"}
    set OpType to ""
    set MyTog to 1
    set Dest to 1
    set StoreText to ""
    set TaskName to ""
    set SubTaskName to ""
    set TaskLimits to {}
    tell application "Finder"
        activate
        --      set MyFol to choose folder with prompt "Which folder contains files
for analysis?"
        set MyFol to ("Macintosh HD:Users:peterwilson:Research:Video
Analysis:TaskAnalysis:Finished:") as alias
        set AllFiles to (every file in MyFol)
    end tell
    ShowMain({0, 0})
    CalcTypes()
    set ARList to {{"Anterior Resection"}, {"Incision", "Laparotomy",
"Retraction", "Mobilise left colon", "Mobilise splenic flexure", "Divide vessels",
"Divide sigmoid", "Mobilise rectum", "Transect rectum", "Prepare colon for
anastomosis", "Anastomosis", "Ileostomy I", "Closure of abdomen", "Ileostomy
II", "Additional procedures", "Dissection in advanced disease"}}
    --  set APList to {"Incision", "Laparotomy", "Retraction", "Mobilise left colon",
"Mobilise splenic flexure", "Divide vessels", "Divide sigmoid", "Mobilise
rectum", "Perineal dissection", "Closure of perineum", "Colostomy I", "Closure
of abdomen", "Colostomy II", "Additional procedures", "Dissection in
advanced disease"}
    set APList to {{"APER"}, {"Incision", "Laparotomy", "Retraction", "Mobilise
left colon", "Mobilise splenic flexure", "Divide vessels", "Divide sigmoid",
"Mobilise rectum", "Perineal dissection", "Closure of perineum", "Ileostomy I",
"Closure of abdomen", "Ileostomy II", "Additional procedures", "Dissection in
advanced disease"}}
    set TempList to {ARList, APList}
```

```
    set TaskList to {}
    repeat with i from 1 to count of OpTypes
        repeat with j from 1 to count of TempList
            if item i of OpTypes contains (item 1 of (item j of TempList)) then copy
(item 2 of (item j of TempList)) to end of TaskList
        end repeat
    end repeat
    CountFiles(0)
end awake from nib

on ShowMain(SH)
    global MyFol
    tell window "Main"
        set visible of text field "NumBox" to (item 1 of SH)
        set visible of text field "NumText" to (item 1 of SH)
        set contents of text field "FolField" to "Analysis of Folder " & (MyFol as
text)
    end tell
end ShowMain

on ShowProg(SH)
    global MyFol
    tell window "ProgWin"
        set contents of text field "FolField" to "Analysis of Folder " & (MyFol as
text)
        set visible to SH
    end tell
end ShowProg


on clicked theObject
    global MyFol, AllFiles, CountFile, TaskLimits, ToEx, OpType, TaskList,
TaskName, MyTog, Dest, FileLimit, StoreText
    set MyB to name of theObject
    set FileLimit to state of button "NumLimit" of window "Main"
    set TaskOn to state of button "TaskLimit" of window "Main"
    set SubTaskOn to state of button "SubTaskLimit" of window "Main"
    set LimitOpOn to state of button "TypeLimit" of window "Main"
    ShowMain({FileLimit})
    if FileLimit = 1 and MyB = "NumLimit" then set UpperLimit to LimitFileNum()
    if MyB contains "TaskLimit" then LimitTask(MyB, TaskOn, SubTaskOn)
    if MyB = "TimeButton" then
        DoTime()
    else if MyB = "ErrButton" then
        DoErr()
    else if MyB = "QuitButton" then
        quit
    else if MyB = "OpenFiles" then
        OpenText()
    else if MyB = "ExFind" or MyB = "InFind" or MyB = "Keyword" then
```

```
        if MyB = "ExFind" then FindEvent("Ex")
        if MyB = "InFind" then FindEvent("In")
        if MyB = "Keyword" then FindEvent("Key")
    else if MyB = "ValidFile" then
        Validate()
    else if MyB = "Exclude" then
        ExTask()
    else if MyB = "Close" then
        set visible of window "ExErr" to 0
    else if MyB = "Add" then
        AddTask()
    else if MyB = "Remove" then
        RemoveTask()
    else if MyB = "FileToggle" then
        set MyTog to current row of matrix "FileToggle" of window "Main"
    else if MyB = "Output" then
        set Dest to current row of matrix "Output" of window "Main"
    else if MyB = "Reset" then
        set ToEx to {"0  0  0  0  0  0  0  0  0", "0    0       1       0       0       0
    0  0  0"}
        ExTask()
    else if MyB = "TypeLimit" then
        LimitOp(LimitOpOn)
    else if MyB = "RedoClip" then
        set the clipboard to StoreText
    end if
end clicked

on LimitFileNum()
    global UpperLimit
    try
        set UpperLimit to (contents of text field "NumBox" of window "Main") as
integer
    on error
        set UpperLimit to 5
    end try
    return UpperLimit
    CountFiles(1)
end LimitFileNum

on CountFiles(FileLimit)
    global AllFiles, CountFile, MyFol, UpperLimit
    tell application "Finder"
        set AllFiles to (every file in MyFol)
        set CountFile to count of items in AllFiles
        if FileLimit = 1 then
            if CountFile is greater than UpperLimit and UpperLimit is greater than
0 then
                set AllFiles to (items 1 through UpperLimit of AllFiles)
                set CountFile to count of items in AllFiles
```

```
            end if
        end if
    end tell
    CalcTypes()
end CountFiles

on LimitTask(MyB, TaskOn, SubTaskOn)
    global OpTypes, TaskLimits, OpType, TaskList, TaskName, SubTaskName
    if MyB = "TaskLimit" then
        if TaskOn = 1 then
            set OpsInUse to {}
            if OpType = "" then
                set ChooseFromOp to (choose from list OpTypes with prompt
"Choose from the following available procedures") as text
            else
                set ChooseFromOp to OpType
            end if
            repeat with i from 1 to count of OpTypes
                if ChooseFromOp contains (item i of OpTypes) then set
ThisOpType to i
            end repeat
            set TaskName to (choose from list (item ThisOpType of TaskList) with
prompt "Please select task") as text
        else
            set TaskName to ""
        end if
        repeat with i from 1 to count of OpTypes
            if (item i of TaskList) contains TaskName then
                repeat with j from 1 to (count of item i of TaskList)
                    if (item j of item i of TaskList) = TaskName then set (item i of
TaskLimits) to j
                end repeat
            else
                set item i of TaskLimits to " "
            end if
            if TaskOn = 0 then set item i of TaskLimits to 0
        end repeat
        set contents of text field "TaskText" of window "Main" to TaskName
    else
        if SubTaskOn = 1 then
            set SubTaskName to (text returned of (display dialog ("Please enter a
subtask to limit search to") default answer ""))
        else
            set SubTaskName to ""
        end if
        set contents of text field "SubTaskText" of window "Main" to
SubTaskName
    end if
end LimitTask
```

```
on CalcTypes()
    global AllFiles, OpTypes, OpType, TaskLimits
    set OpTypes to {}
    tell application "Finder"
        repeat with i from 1 to count of AllFiles
            set AppleScript's text item delimiters to "OpType: "
            set ThisText to read (((item i of AllFiles) as text) as alias)
            if (count of text items in ThisText) is greater than 1 then
                set ThisType to paragraph 1 of (text item 2 of ThisText)
                set AppleScript's text item delimiters to tab
                set ThisType to text item 1 of ThisType
                if OpTypes does not contain ThisType then copy ThisType to end
of OpTypes
            end if
        end repeat
    end tell
    if TaskLimits = {} then
        repeat with i from 1 to count of OpTypes
            copy 0 to end of TaskLimits
        end repeat
    end if
end CalcTypes

on LimitOp(LimitOpOn)
    global AllFiles, OpTypes, OpType, TaskLimits
    if LimitOpOn = 1 then
        set ChooseType to {}
        repeat with i from 1 to count of TaskLimits
            if item i of TaskLimits is not equal to " " then copy item i of OpTypes to
end of ChooseType
        end repeat
        set OpType to (choose from list ChooseType) as text
    else
        set OpType to ""
    end if
    set contents of text field "TypeText" of window "Main" to OpType
end LimitOp


on RemoveTask()
    global ToEx
    set RemText to contents of combo box "Combo" of window "ExErr"
    set TempEx to {}
    repeat with e from 1 to count of ToEx
        copy item e of ToEx to end of TempEx
    end repeat
    set ToEx to {}
    repeat with e from 1 to count of TempEx
        if item e of TempEx is not equal to RemText then copy item e of TempEx
to end of ToEx
```

```
    end repeat
    ExTask()
end RemoveTask

on ExTask()
    global ToEx
    set visible of window "ExErr" to 1
    set ExText to ""
    repeat with e from 2 to count of ToEx
        set ExText to ExText & (item e of ToEx) & return
    end repeat
    set contents of text field "ExList" of window "ExErr" to ExText
    delete every combo box item of combo box "combo" of window "ExErr"
    repeat with e from 1 to count of paragraphs of ExText
        make new combo box item at end of combo box items of combo box
"Combo" of window "ExErr" with data (paragraph e of ExText)
    end repeat
end ExTask

on AddTask()
    global ToEx
    set NewEx to ""
    tell window "ExErr"
        repeat with e from 1 to 8
            if e = 8 then
                set NewEx to NewEx & "0"
                exit repeat
            end if
            set ThisBox to contents of text field ("Box" & e)
            set NewEx to NewEx & ThisBox & tab
            if e = 4 then set NewEx to NewEx & ThisBox & tab
        end repeat
    end tell
    if ToEx does not contain NewEx then copy NewEx to end of ToEx
    ExTask()
end AddTask

on Validate()
    global MyFol, AllFiles, ErrorList
    set AllErr to {}
    set MyPath to MyFol as text
    set AppleScript's text item delimiters to ":"
    set MyPath to (text items 1 thru -3 of MyPath) as text
    set ErrPath to MyPath & ":ErrorDocuments"
    --  display dialog "Path = " & ErrPath
    set AppleScript's text item delimiters to tab
    tell application "Finder"
        repeat with i from 1 to (count of ErrorList) - 1
            copy {} to end of AllErr
            set ThisDoc to ErrPath & ":" & (item i of ErrorList)
```

```
            set ThisText to read (ThisDoc as alias)
            repeat with j from 2 to count of paragraphs of ThisText
               set ThisLine to every text item of (paragraph j of ThisText)
               if (count of ThisLine) is greater than 1 then copy (item -1 of
ThisLine) to end of item -1 of AllErr
            end repeat
         end repeat
      end tell
      ShowProg(1)
      --      display dialog "There are " & (count of AllFiles) & " items to analyse"
      set CountFile to count of AllFiles
      -- set OldNames to {"EEM Mode", "Error type", "Recovery mechanism",
"Failure to", "Preparatory measure"}
      repeat with i from 1 to count of AllFiles
         set MyN to name of item i of AllFiles
         tell window "ProgWin" to set contents of text field "TexField" to "Analysis
of Text File " & (MyN as text)
         tell window "ProgWin" to tell progress indicator "FolProg" to set contents
to (i / CountFile) * 100
         --          display dialog "Analysing item " & i & " = " & (item i of AllFiles)
         tell application "Finder" to set ThisText to read ((item i of AllFiles as text)
as alias)
         set CountP to count of paragraphs in ThisText
         repeat with j from 1 to CountP
            set DoQuit to false
            set ThisP to paragraph j of ThisText
            set AppleScript's text item delimiters to ":"
            set CheckErr to every text item in ThisP
            if (count of items in CheckErr) is greater than 2 and ThisP does not
contain "Time of Analysis" and ThisP does not contain "Instrument change"
then
               set ErrType to item 2 of CheckErr
               set InList to false
               repeat with k from 1 to count of ErrorList
                  if ErrType contains "Type of " & (item k of ErrorList) then
                     try
                        set ErrNum to (word 1 of item 3 of CheckErr) as integer
                        set OrigErr to item ErrNum of item k of AllErr
                        set AppleScript's text item delimiters to ". "
                        set ErrText to (text items 2 thru -1 of (item 3 of CheckErr))
as text
                     on error
                        set ErrNum to 0
                        set OrigErr to "[Not defined]"
                        set ErrText to (item 3 of CheckErr)
                     end try
                     if OrigErr is not equal to ErrText then
                        display dialog "Item " & ErrNum & " of " & (item k of
ErrorList) & " is " & return & ErrText & return & "It should be " default answer
OrigErr buttons {"OK", "Cancel", "Choose"} default button "OK"
```

```
                    if button returned of result = "Choose" then
                        set NewItem to choose from list (item k of AllErr)
                        repeat with z from 1 to count of (item k of AllErr)
                            if (NewItem as text) = ((item z of item k of AllErr) as
text) then
                                set ListPos to z
                                exit repeat
                            end if
                        end repeat
                        display dialog "Use following text: " default answer ("" &
ListPos & ". " & NewItem & ": ") buttons {"OK", "Cancel"}
                        set Entry to result
                        if button returned of Entry = "OK" then set ThisText to
DoText(MyN, ThisText, ThisP, text returned of result)

                    end if
                end if
                set InList to true
                exit repeat
            end if
        end repeat
        if InList = false then
            tell application "TextEdit" to open (item i of AllFiles)
            display dialog "Please check paragraph " & j & " of " & MyN & "
as " & ErrType & " is not in the list" & return & ThisP buttons {"OK", "Quit"}
            if button returned of result = "Quit" then set DoQuit to true
        end if
    end if
    if DoQuit = true then exit repeat
end repeat


end repeat
tell window "ProgWin" to set visible to false
end Validate

on DoText(MyN, ThisText, ThisP, NewText)
    global MyFol
    tell application "TextEdit"
        if name of window 1 does not contain MyN then open (MyFol & MyN) as
text
        set AppleScript's text item delimiters to ThisP
        set OldText to every text item in ThisText
        set AppleScript's text item delimiters to ": "
        set OldP to {text item 1 of ThisP, text items 2 thru -1 of ThisP}
        set OldP1 to (item 1 of OldP) as text
        set OldP2 to (item 2 of OldP) as text
        set NewP to OldP1 & ": " & NewText & OldP2
        display dialog "New Para = " & NewP
```

```
        if button returned of result = "OK" then set ThisText to "" & (item 1 of
OldText) & NewP & (item 2 of OldText)
        set text of document 1 to ThisText
    end tell
    return ThisText
end DoText

on ReadFiles()
    global FileText, ErrorList, MyTog, AllFiles, OpType, TaskLimits, TaskList,
OpTypes, AllText, MyN
    set AllText to {}
    set MyN to {}
    if MyTog = 2 then
        display dialog "Do you wish to Choose filed, Type names of files, or Use
open files?" buttons {"Choose", "Type", "Use"} default button "Type"
        set FindButton to button returned of result
    else
        set ToOpen to {}
        repeat with i from 1 to count of AllFiles
            copy (item i of AllFiles) as text to end of ToOpen
        end repeat
        set FindButton to "No need to Type"
    end if
    if FindButton = "Choose" then
        tell application "Finder"
            set ChosenFiles to choose file with multiple selections allowed
            repeat with ThisFile in ChosenFiles
                copy name of ThisFile to end of MyN
                set ThisText to read (ThisFile)
                copy ThisText to end of AllText
            end repeat
        end tell
    else if FindButton contains "Type" then
        if FindButton = "Type" then set ToOpen to FileList()
        tell application "Finder"
            set AppleScript's text item delimiters to ":"
            repeat with i from 1 to count of ToOpen
                try
                    set ThisFile to (item i of ToOpen)
                    set ThisName to text item -1 of ThisFile
                    copy ThisName to end of MyN
                    set ThisText to read (ThisFile as alias)
                    copy ThisText to end of AllText
                on error
                    display dialog "Could not read (" & ThisFile & ")"
                end try
            end repeat
        end tell
    else
        tell application "TextEdit"
```

```
            set TexWin to name of every window
            set AppleScript's text item delimiters to "  —  /"
            repeat with i from 1 to count of TexWin
                copy (text of document i) to end of AllText
                copy (text item 1 of (item i of TexWin)) to end of MyN
            end repeat
        end tell
    end if
end ReadFiles

on FindEvent(ExIn)
    global FileText, ErrorList, MyTog, AllFiles, OpType, TaskLimits, TaskList,
OpTypes, AllText, MyN
    set FindList to {}
    set FileText to ""
    ReadFiles()
    if ExIn = "Key" then
        display dialog "Search with AND or with OR?" buttons {"AND", "OR"}
default button "AND"
        set KeyBool to button returned of result
        if KeyBool = "AND" then
            set ExIn to "Ex"
        else
            set ExIn to "In"
        end if
        repeat
            display dialog "Enter keyword to search for (blank to exit):" default
answer ""
            set FindWord to text returned of result
            if FindWord = "" then exit repeat
            copy FindWord to end of FindList
        end repeat
    else
        repeat
            set FindErr to "Type of " & (choose from list ErrorList with prompt
"Which error type to find?") & ": "
            display dialog "Which Error Number to find for " & FindErr & return &
"Leave blank to find all " & FindErr default answer ""
            set ErrNum to text returned of result
            if ErrNum is not equal to "" then set FindErr to (FindErr & ErrNum &
".")
            copy FindErr to end of FindList
            display dialog "Add Another search criterion, or Exit?" buttons
{"Another", "Exit"} default button "Another"
            if button returned of result = "Exit" then exit repeat
        end repeat
    end if
    set AppleScript's text item delimiters to "."
    ShowProg(1)
    repeat with i from 1 to count of AllText
```

```
        tell window "ProgWin" to set contents of text field "TexField" to "Analysis
of Text File " & (item i of MyN as text)
        tell window "ProgWin" to tell progress indicator "FolProg" to set contents
to (i / (count of AllText)) * 100
        set ThisText to item i of AllText
        if OpType = "" or ThisText contains "OpType: " & OpType then
            set AppleScript's text item delimiters to "OpType: "
            set ThisType to paragraph 1 of text item 2 of ThisText
            set AppleScript's text item delimiters to tab
            set ThisType to text item 1 of ThisType
            set AppleScript's text item delimiters to "."
            set OpPos to 1
            repeat with a from 1 to count of OpTypes
                if ThisType = (item a of OpTypes) then set OpPos to a
            end repeat
            set TaskLimit to (item OpPos of TaskLimits)
            set CountP to count of paragraphs of ThisText
            repeat with j from 1 to CountP
                if (j mod 5 = 0) then tell window "ProgWin" to tell progress indicator
"TexProg" to set contents to (j / CountP) * 100
                set ThisP to paragraph j of ThisText
                set TempText to ""
                try
                    if ExIn = "In" then
                        repeat with f from 1 to count of FindList
                            set FindErr to item f of FindList
                            if ThisP contains FindErr then
                                try
                                    set Time1 to word -1 of ThisP
                                    set T to TaskLimit
                                    if TaskLimit is not equal to 0 then set T to ((word -1 of
text item 1 of ThisP) as real)
                                    if T = TaskLimit then
                                        repeat with k from (j - 4) to (j + 4)
                                            if k is greater than (count of paragraphs in
ThisText) then exit repeat
                                            set NearPara to paragraph k of ThisText
                                            if NearPara is not equal to "" then
                                                set Time2 to word -1 of NearPara
                                                if Time2 = Time1 then set FileText to FileText
& (item i of MyN) & tab & NearPara & tab & (((Time1 as real) as integer) /
(3600 * 24)) & return
                                            end if
                                        end repeat
                                    end if
                                on error
                                    display dialog "Problem with " & (item i of MyN) & " at
" & ThisP & " " & NearPara
                                end try
                            end if
```

```
                end repeat
            else
                if ThisP contains (item 1 of FindList) then
                    set Time1 to word -1 of ThisP
                    set NearList to {}
                    repeat with M from (j - 4) to (j + 4)
                        if M is greater than (count of paragraphs in ThisText)
then exit repeat
                        set NearPara to paragraph M of ThisText
                        if NearPara is not equal to "" then
                            if word -1 of NearPara = Time1 then copy NearPara
to end of NearList
                        end if
                    end repeat
                    set AllPresent to 0
                    repeat with M from 1 to count of NearList
                        set ThisPara to item M of NearList
                        repeat with n from 1 to count of FindList
                            set ToFind to item n of FindList
                            if ThisPara contains ToFind then set AllPresent to
AllPresent + 1
                        end repeat
                    end repeat
                    if AllPresent = (count of FindList) then
                        try
                            set T to TaskLimit
                            set LookTask to ""
                            if TaskLimit is not equal to 0 then
                                set T to ((word -1 of text item 1 of ThisP) as
integer)
                                set LookTask to (item T of item OpPos of TaskList)
                            end if
                            if T = TaskLimit then
                                --if (count of paragraphs in FileText) mod 25 = 0
then display dialog "Task " & T & " of " & ThisType & " should be " &
LookTask giving up after 1
                                repeat with M from 1 to count of NearList
                                    set NearPara to item M of NearList
                                    set AddText to (item i of MyN) & tab & NearPara
& tab & (((Time1 as real) as integer) / (3600 * 24)) & return
                                    set FileText to FileText & AddText
                                end repeat
                            end if
                        on error
                            display dialog "Problem at middle loop of FindEvent in
" & (item i of MyN) & " at " & ThisP
                        end try
                    end if
                end if
            end if
```

```
            on error
                display dialog "Problem with outer loop of FindEvent in " & (item
i of MyN) & " at " & ThisP & " " & NearPara
            end try
        end repeat
    end if
  end repeat
  tell window "ProgWin" to set visible to false
  WriteText(FileText, 1)
end FindEvent

on FileList()
  global MyFol
  display dialog "Write or paste procedures to open here:" default answer ""
  set Temp0 to text returned of result
  set Temp1 to ""
  repeat with p from 1 to count of paragraphs in Temp0
    set Temp1 to Temp1 & (paragraph p of Temp0)
  end repeat
  set AppleScript's text item delimiters to "Op "
  set Temp1 to every text item in Temp1
  set AppleScript's text item delimiters to space
  set TextFiles to {}
  repeat with T from 1 to count of Temp1
    set Temp1Item to item T of Temp1
    set Temp2 to every text item of Temp1Item
    repeat with U from 1 to count of Temp2
      set OpShortName to item U of Temp2
      if length of OpShortName = 2 then set OpShortName to ("0" &
OpShortName)
      set OpNum to "Op " & (OpShortName)
      if OpNum is not equal to "Op " and TextFiles does not contain OpNum
then copy OpNum to end of TextFiles
    end repeat
  end repeat
  set AppleScript's text item delimiters to ":"
  set ToOpen to {}
  repeat with i from 1 to count of TextFiles
    try
      set ThisFile to (item i of TextFiles)
      --          if ThisFile does not contain "Op " then set ThisFile to "Op "
& ThisFile
      set ThisFile to (MyFol as text) & ThisFile
      copy ThisFile to end of ToOpen
    on error
      display dialog "Could not open file (" & ThisFile & ")"
    end try
  end repeat
  return ToOpen
end FileList
```

```
on OpenText()
    global MyFol
    set ToOpen to FileList()
    tell application "TextEdit"
        activate
        repeat with i from 1 to count of ToOpen
            --display dialog "Will be opening " & (item i of ToOpen)
            try
                open item i of ToOpen
                --          on error
            end try
        end repeat
    end tell
end OpenText

on WriteText(FileText, RidTabs)
    global Dest, StoreText
    if RidTabs = 1 then
        repeat 3 times
            set AppleScript's text item delimiters to "                    "
            set NewText to every text item in FileText
            set AppleScript's text item delimiters to "          "
            set FileText to (NewText as text)
            if FileText does not contain "      " then exit repeat
        end repeat
    end if
    if Dest = 1 then
        set the clipboard to FileText
        tell application "Microsoft Excel" to Activate
    else
        tell application "TextEdit"
            launch
            make new document at beginning
            set text of document 1 to FileText
            activate
        end tell
    end if
    set StoreText to FileText
end WriteText

on DoErr()
    global MyFol, AllFiles, CountFile, AllTasks, AllRef, CountList, CRef, ThisP,
PRef, TaskLimits, ErrorList, ToEx, OpType, OpTypes, TaskName, MyN,
AllText, SubTaskName
    set ErrorList2 to {"EEM", "Failure", "Error", "Consequence", "Cons2",
"Recovery", "Preparation", "Techniques", "Other"}
    ReadFiles()
    ShowProg(1)
    --  set FRef to a reference to AllFiles
```

```
    --  set AllList to {{}, {}, {}, {}, {}, {}}
    set AllTasks to {}
    set AllRef to a reference to AllTasks
    set CountList to {}
    set CRef to a reference to CountList
    set ML to {}
    set FL to {}
    set EL to {}
    set CL to {}
    set RL to {}
    set TL to {}
    set OLBE to {}
    set ErrPerOp to {}
    set AddT to ""
    set OpList to {}
    --set DoDial to true
    set AppleScript's text item delimiters to ":   Type of"
    set TimeA to time of (current date)
    repeat with i from 1 to count of AllText
        set ThisName to item i of MyN
        tell window "ProgWin" to set contents of text field "TexField" to "Analysis
of Text File " & ThisName
        tell window "ProgWin" to tell progress indicator "FolProg" to set contents
to (i / (count of AllText)) * 100
        set ThisText to item i of AllText
        --  repeat with i from 1 to CountFile
        --      set ThisD to (item i of FRef) as text
        --      set MyN to name of item i of FRef
        --      tell window "ProgWin" to set contents of text field "TexField" to
"Analysis of Text File " & (MyN as text)
        --      tell window "ProgWin" to tell progress indicator "FolProg" to set
contents to (i / CountFile) * 100
        --      set ThisText to read (ThisD as alias)
        if OpType = "" or ThisText contains "OpType: " & OpType then
            set AppleScript's text item delimiters to "OpType: "
            set ThisType to paragraph 1 of text item 2 of ThisText
            set AppleScript's text item delimiters to tab
            set ThisType to text item 1 of ThisType
            set TaskLimit to 99
            repeat with OT from 1 to count of OpTypes
                if ThisType = item OT of OpTypes then
                    set OpNum to OT
                    set TaskLimit to item OpNum of TaskLimits
                    --display dialog "This operation = " & ThisType & " which is
number " & OpNum & "; TaskLimit = " & TaskLimit
                end if
            end repeat
            if TaskLimit is not equal to " " then
                set ThisP to every paragraph in ThisText
                set CountP to count of ThisP
```

```
            set PRef to (a reference to ThisP)
            copy {"", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""} to end of
ErrPerOp
            set AddL to {0, 0, 0, 0, 0, 0, 0, 0, 0}
            set ErrInOp to 0
            --      if MyN = "Op 106" then
            repeat with j from 1 to CountP
               tell window "ProgWin" to tell progress indicator "TexProg" to set
contents to (j / CountP) * 100
               set ThisL to item j of PRef
               try
                  set AppleScript's text item delimiters to "."
                  set ThisTask to ((word -1 of text item 1) of ThisL)
                  set ThisTask to ThisTask as integer
                  if (item ThisTask of item -1 of ErrPerOp) = "" then set (item
ThisTask of item -1 of ErrPerOp) to 0
               end try
               if TaskLimit = 0 or TaskLimit = ThisTask then
                  if SubTaskName = "" or ThisL contains ("." & SubTaskName
& ".") then
                     if ThisL contains ": Type of" then
                        set AppleScript's text item delimiters to ":        Type of"
                        set W1 to word 1 of text item 2 of ThisL
                        set W2 to word 2 of text item 2 of ThisL
                        try
                           try
                              set W2 to W2 as integer
                              set W3 to W2
                           on error
                              set W3 to character 1 of W2
                           end try
                           set InList to false
                           repeat with k from 1 to count of ErrorList2
                              if W1 = (item k of ErrorList2) then
                                 set item k of AddL to W2
                                 set InList to true
                                 if k = 4 then set item 5 of AddL to W3
                                 --display dialog "OldTime = " & OldTime & return
& "AddT = " & AddT
                                 --display dialog "ThisLine = " & ThisL & return &
"OldTime = " & OldTime & return & "AddT = " & AddT & return & "W1 = " &
W1 & "; W2 = " & W2
                              end if
                           end repeat
                           if InList = false then set item -1 of AddL to W2
                        end try
                        set NewTime to (word -1 of ThisL)
                        try
                           set NextTime to (word -1 of (item (j + 1) of PRef))
                        on error
```

```
                        -- display dialog "Could not set NextTime"
                        set NextTime to NewTime & " "
                    end try
                    if NewTime is not equal to NextTime then
                        set AppleScript's text item delimiters to "."
                        set NewTime to (text item 1 of NewTime) as integer
                        -- display dialog "Found error at new time in " & MyN
& return & ThisL
                        set AddT to ""
                        repeat with a from 1 to count of AddL
                            set AddT to AddT & (item a of AddL)
                            if a = (count of AddL) then exit repeat
                            set AddT to AddT & ""
                        end repeat
                        set NotEx to true
                        set whichEx to "NA"
                        repeat with e from 1 to count of ToEx
                            set ThisEx to (item e of ToEx)
                            if AddT = ThisEx then
                                set NotEx to false
                                set whichEx to ThisEx
                            end if
                            --      if DoDial = true then
                            --          display dialog "Checking if " & tab & "(" &
ThisEx & ")" & return & "is equal to " & tab & "(" & AddT & ")" buttons {"OK",
"Exit"} default button "OK"
                            --          if button returned of result = "Exit" then set
DoDial to false
                            --      end if
                        end repeat
                        set AddL to {0, 0, 0, 0, 0, 0, 0, 0, 0}
                        try
                            if NotEx = true then
                                set ErrInOp to ErrInOp + 1
                                set AppleScript's text item delimiters to "."
                                set ThisTask to ((word -1 of text item 1) of ThisL)
                                set AppleScript's text item delimiters to ":
    Type of"
                                set ThisTask to ThisTask as integer
                                --          if (item ThisTask of item -1 of ErrPerOp)
= "" then set (item ThisTask of item -1 of ErrPerOp) to 0
                                set (item ThisTask of item -1 of ErrPerOp) to
(item ThisTask of item -1 of ErrPerOp) + 1
                                --      if DoDial = true then
                                --          display dialog "Adding  E       F       Err
    C  C2    R  P  T" & return & tab & tab & AddT & return & " to task " &
ThisTask buttons {"OK", "Exit"} default button "OK"
                                --          if button returned of result = "Exit" then
set DoDial to false
                                --      end if
```

```
                                else
                                    --      display dialog "      E       F       P       Err
    C  T  T" & return & tab & AddT & "(" & whichEx & ") is excluded from task "
& ThisTask
                                end if
                            end try
                            if AllRef contains AddT then
                                repeat with n from 1 to count of AllTasks
                                    if (item n of AllRef) = AddT then
                                        set (item n of CRef) to ((item n of CRef) + 1)
                                        try
                                            if item n of OLBE does not contain
ThisName then set item n of OLBE to (item n of OLBE) & " " & ThisName
                                            --on error
                                            --display dialog "Could not add " & MyN & "
to item " & n & " of " & (OLBE as text)
                                        end try
                                        exit repeat
                                    end if
                                end repeat
                            else
                                copy AddT to end of AllRef
                                copy 1 to end of CRef
                                copy ThisName to end of OLBE
                            end if
                        end if
                    end if
                end if
            end if
        end repeat
        set AppleScript's text item delimiters to tab
        --copy (text item 1 of item 1 of ThisP) &
        copy MyN & tab & ErrInOp to end of OpList
    end if
    set AppleScript's text item delimiters to ": Type of"
  end if
end repeat
set AppleScript's text item delimiters to ""
set TimeB to time of (current date)
set TimeD to TimeB - TimeA
--return CountList
set TotalFreq to 0
set E1Freq to 0
set TotalErr to 0
set FileText to "EEM Failure Error Consequence    Cons2 Recovery
Preparation  Techniques  Other   Error Event       Rec or Err
Frequency" & return
repeat with i from 1 to (count of AllTasks)
  try
    set ThisFreq to item i of CRef
```

```
            set TotalFreq to TotalFreq + ThisFreq
            set ErrEv to (word 1 of item i of AllRef) + (word 2 of item i of AllRef) +
(word 3 of item i of AllRef) + (word 5 of item i of AllRef)
            if ErrEv is not equal to 0 then set ErrEv to 1
            set RecErr to ErrEv + (word 6 of item i of AllRef)
            if RecErr is not equal to 0 then set RecErr to 1
            if RecErr = 1 then set TotalErr to TotalErr + ThisFreq
            if (word 3 of item i of AllRef) = "1" then set E1Freq to E1Freq +
ThisFreq
            set FileText to FileText & (item i of AllRef) & tab & ErrEv & tab &
RecErr & tab & ThisFreq
            --          try
            set FileText to FileText & tab & (item i of OLBE)
            --          on error
            --              display dialog "Problem adding (" & (item i of OLBE)
            --          end try
            set FileText to FileText & return
        end try
    end repeat
    set FileText to FileText & "                                        Non E1 Errors"
& tab & (TotalErr - E1Freq) & return
    set FileText to FileText & "                                        Error Events" &
tab & TotalErr & return
    set FileText to FileText & "                                        Total" & tab &
TotalFreq
    set TabRight to "                                                   "
    set FileText to FileText & return & TabRight
    set FileText to FileText & TabRight & "Analysis of task types for folder " &
(MyFol as text) & return
    set FileText to FileText & TabRight & "Analysed in " & TimeD & " secs" &
return
    set SubArray to {}
    if TaskName is not equal to "" then copy "Task " & TaskName to end of
SubArray
    if OpType is not equal to "" then copy ("OpType " & OpType) to end of
SubArray
    if SubArray is not equal to {} then
        set FileText to FileText & TabRight & "Sub-analysis limited to "
        repeat with i from 1 to count of SubArray
            set FileText to FileText & (item i of SubArray)
            if i = (count of SubArray) then exit repeat
            set FileText to FileText & "; "
        end repeat
        set FileText to FileText & return
    end if
    set AddTab to "                                                 "
    set FileText to FileText & AddTab -- & tab & "Total"
    repeat with i from 1 to 16
        set FileText to FileText & tab & "Task " & i
    end repeat
```

```
    set FileText to FileText & tab & "Total"
    repeat with i from 1 to count of OpList
        set FileText to FileText & return & AddTab & (item i of MyN)
        set TotalForOp to 0
        repeat with j from 1 to 16
            set ThisErrThisOp to (item j of item i of ErrPerOp)
            set FileText to FileText & tab & ThisErrThisOp
            set TotalForOp to TotalForOp + ThisErrThisOp
        end repeat
        set FileText to FileText & tab & TotalForOp
    end repeat
    tell window "ProgWin" to set visible to false
    WriteText(FileText, 0)
    return CountList
end DoErr

on DoTime()
    global MyFol, AllFiles, CountFile, TaskLimit, TaskLimits, AllTasks, AllRef,
CountList, CRef, ThisP, PRef, ErrorList, ToEx, OpType, OpTypes, TaskName,
MyN, AllText
    ReadFiles()
    ShowProg(1)
    set ActList to {"Delay and No View", "Errors in Each Task"}
    set OpList to {}
    set OpTime to {}
    set TotList to {}
    set DelayList to {}
    set E1List to {}
    repeat 16 times
        copy {} to end of TotList
        copy {} to end of DelayList
        copy {} to end of E1List
    end repeat
    set StartTime to time of (current date)
    repeat with a from 1 to count of AllText
        tell window "ProgWin" to set contents of text field "TexField" to "Analysis
of Text File " & (item a of MyN as text)
        tell window "ProgWin" to tell progress indicator "FolProg" to set contents
to (a / (count of AllText)) * 100
        set ThisText to item a of AllText
        --  repeat with a from 1 to CountFile
        --      tell window "ProgWin" to tell progress indicator "FolProg" to set
contents to (a / CountFile) * 100
        --      try
        --      tell application "Finder"
        --      set ThisItem to (item a of AllFiles) as alias
        --      set MyN to name of ThisItem
        --      set ThisText to read ThisItem
        --  end tell
```

```
	-- tell window "ProgWin" to set contents of text field "TexField" to
"Analysis of Text File " & (MyN as text)
	--	end try
	if OpType = "" or ThisText contains "OpType: " & OpType then
		set AppleScript's text item delimiters to "OpType: "
		set ThisType to paragraph 1 of text item 2 of ThisText
		set AppleScript's text item delimiters to tab
		set ThisType to text item 1 of ThisType
		set TaskLimit to 99
		repeat with OT from 1 to count of OpTypes
			if ThisType = item OT of OpTypes then
				set OpNum to OT
				set TaskLimit to item OpNum of TaskLimits
				--display dialog "This operation = " & ThisType & " which is
number " & OpNum & "; TaskLimit = " & TaskLimit
			end if
		end repeat
		if TaskLimit is not equal to " " then
			copy (item a of MyN) to end of OpList
			set FileAsPara to (every paragraph in ThisText)
			set ParaCount to (count of items in FileAsPara)
			repeat with b from 1 to 16
				copy 0 to end of item b of TotList
				copy 0 to end of item b of DelayList
				copy 0 to end of item b of E1List
			end repeat
			set MaxTime to 0
			set MinTime to 1000
			set AppleScript's text item delimiters to "."
			repeat with i from 3 to ParaCount - 1
				if i mod 5 = 0 then tell window "ProgWin" to tell progress
indicator "TexProg" to set contents to (i / ParaCount) * 100
				try
					set ThisEntry to (item i of FileAsPara)
					if ThisEntry does not contain "Date & Time" and ThisEntry
does not contain "TimeElapsed" and ThisEntry contains ":	" then
						repeat with j from 1 to 10
							set NextEntry to (item (i + j) of FileAsPara)
							if NextEntry contains ":  " and NextEntry does not
contain "Date & Time" and NextEntry does not contain "TimeElapsed" then
exit repeat
						end repeat
						set T to (word -1 of text item 1 of ThisEntry) as integer
						set TimeA to last word of (ThisEntry) as integer
						if TimeA is less than MinTime then set MinTime to TimeA
						if TimeA is greater than MaxTime then set MaxTime to
TimeA
						set TimeB to last word of (NextEntry) as integer
						set TimeDiff to (TimeB - TimeA)
						if TaskLimit = 0 or TaskLimit = T then
```

```
                        if TimeDiff is greater than 0 then
                            set (item -1 of item T of TotList) to (item -1 of item T
of TotList) + TimeDiff
                            if ThisEntry contains "Delay" then
                                set (item -1 of item T of DelayList) to (item -1 of
item T of DelayList) + TimeDiff
                            else if ThisEntry contains "Poor camera views" then
                                set (item -1 of item T of E1List) to (item -1 of item T
of E1List) + TimeDiff
                            end if
                        else if TimeDiff is less than -20 then
                            --                  display dialog "This is task " & (word
1 of ThisEntry) & " (Task " & T & "). Will add " & TimeDiff & "." buttons {"OK",
"Quit"} default button "OK"
                            --                  if button returned of result = "Quit"
then exit repeat

                        end if
                    end if
                end if
                --  on error
                --      if i mod 5 = 0 then display dialog "Error with operation
" & MyN & " reading task " & T & " at line " & i giving up after 2
            end try
        end repeat
        copy (MaxTime - MinTime) to end of OpTime
    end if
  end if
end repeat
set FileText to " Total                    "
repeat with a from 1 to 16
  set FileText to FileText & "Task " & a & "                                    "
end repeat
set FileText to FileText & return & "Name of File     Time   hh:mm:ss     "
repeat with a from 1 to 16
  set FileText to FileText & "Delay       No View                    "
end repeat
set EndTime to time of (current date)
set FileText to FileText & return
repeat with i from 1 to count of OpList
  --    try
  set FileText to FileText & (item i of OpList) & tab
  set TotDel to 0
  set TotE1 to 0

  set TempText to ""
  repeat with a from 1 to 16
    set ThisTot to (item i of item a of TotList)
    set ThisDel to (item i of item a of DelayList)
    set ThisE1 to (item i of item a of E1List)
```

```
            if ThisTot is greater than 0 then
                set TempText to TempText & "" & ThisTot & tab & ThisDel & tab &
(ThisDel / ThisTot) & tab & ThisE1 & tab & (ThisE1 / ThisTot) & tab
            else
                set TempText to TempText & "" & tab & tab & tab & tab & tab
            end if
            set TotDel to TotDel + ThisDel
            set TotE1 to TotE1 + ThisE1
        end repeat
        set TotTime to item i of OpTime
        set FileText to FileText & TotTime & tab & (TotTime / (3600 * 24)) & tab
& TotDel & tab & (TotDel / TotTime) & tab & TotE1 & tab & (TotE1 / TotTime)
& tab & TempText & return
        --      end try
    end repeat
    tell window "ProgWin" to set visible to false
    WriteText(FileText, 0)
    --  set NewFile to choose file name with prompt "Choose somewhere to
save this analysis file"
    --  set WriteFile to (open for access NewFile with write permission)
    --  write "Analysis of folder " & MyFol & " on date " & (current date) & return
& return & FileText to WriteFile starting at eof
    --  close access NewFile
    --  display dialog "That method took " & (EndTime - StartTime) & " secs"
end DoTime
```