A \LaTeX\ macro package for Message Sequence Charts

Reference Manual

Victor Bos  Ton van Deursen  Sjouke Mauw
 Universit"e du Luxembourg  Universit"e du Luxembourg
vbos@abo.fi  ton.vandeursen@uni.lu  sjouke.mauw@uni.lu

Version 1.17, last update December 22, 2010
Describing msc macro package version 1.17

Contents
1 Introduction 1
2 Concepts 1
3 Environments 3
4 Commands 6
5 User definable lengths 12
6 Internal lengths 14
7 Internal boxes 15
8 Internal counters 15
9 Limitations 15
10 Tricks 15

1 Introduction

The msc macro package was developed to draw (actually, to write) Message Sequence Charts (MSCs) with \LaTeX. The current version supports most of the MSC language standardized in [2]. The manual [1] describes how to use the msc macro package and is illustrated with numerous examples. This reference manual briefly describes the main concepts of the package and it provides lists of all available environments and commands. In addition, it lists both the user-definable lengths and the internal lengths that are used by the package to compute the layout of MSCs.

2 Concepts

The msc macro package offers three different kinds of diagrams:

- MSC diagrams (normal MSCs)
- HMSC diagrams (high level MSCs)
MSC diagrams (MSC documents)

For each of these diagrams, the package provides a \texttt{LATEX} environment. Figure 1 shows an example of each diagram. The source code for these diagrams is given in Figure 2. Depending on the environment, different MSC macro package commands can be used. Furthermore, since each environment is implemented as a pspicture (see pstricks documentation), it is possible to use pstricks commands inside the MSC environments.

\begin{figure}[h]
\centering
\begin{tabular}{lll}
\textbf{MSC Example 1} & \textbf{HMSC Example 2} & \textbf{MSCdoc Example 3} \\
\begin{minipage}{0.3\textwidth}
\begin{verbatim}
\begin{msc}
  {Example 1}
  \declinst{i}{$i$}{}
  \declinst{j}{$j$}{}
  \nextlevel
  \mess{a}{i}{j}
  \nextlevel[2]
  \mess{b}{j}{i}
\end{msc}
\end{verbatim}
\end{minipage} & \begin{minipage}{0.3\textwidth}
\begin{verbatim}
\begin{hmsc}
  {Example 2}
  (0.0)(4,4.65)
  \hmscstartsymbol{S}(2,3.75)
  \hmscconnection{c}(2,3.25)
  \hmscreference{R1}{A}(2,2.5)
  \hmsccondition{C}{?}(2,1.5)
  \hmscendsymbol{E}(1.5,0.5)
  \arrow{S}{c}
  \arrow{c}{R1}
  \arrow{R1}{C}
  \arrow{C}
  \{(2.5,1)(3,1)(3,3.25)]
  \{(c)
  \arrow{C}[,(1.5,1)]{E}
\end{hmsc}
\end{verbatim}
\end{minipage} & \begin{minipage}{0.3\textwidth}
\begin{verbatim}
\begin{mscdoc}
  {Example 3}
  (0,0)(4,4.65)
  \reference{A}(1,3.0)
  \reference{B}(3,3.0 )
  \separator{2.0}
  \reference{C}(2,1.0)
\end{mscdoc}
\end{verbatim}
\end{minipage}
\end{tabular}
\caption{Examples of different diagrams}
\end{figure}

\begin{figure}[h]
\centering
\begin{minipage}{\textwidth}
\begin{verbatim}
\begin{msc}
  {Example 1}
  \declinst{i}{$i$}{}
  \declinst{j}{$j$}{}
  \nextlevel
  \mess{a}{i}{j}
  \nextlevel[2]
  \mess{b}{j}{i}
\end{msc}
\end{verbatim}
\begin{verbatim}
\begin{hmsc}
  {Example 2}
  (0,0)(4,4.65)
  \hmscstartsymbol{S}(2,3.75)
  \hmscconnection{c}(2,3.25)
  \hmscreference{R1}{A}(2,2.5)
  \hmsccondition{C}{?}(2,1.5)
  \hmscendsymbol{E}(1.5,0.5)
  \arrow{S}{c}
  \arrow{c}{R1}
  \arrow{R1}{C}
  \arrow{C}
  \{(2.5,1)(3,1)(3,3.25)]
  \{(c)
  \arrow{C}[,(1.5,1)]{E}
\end{hmsc}
\end{verbatim}
\begin{verbatim}
\begin{mscdoc}
  {Example 3}
  (0,0)(4,4.65)
  \reference{A}(1,3.0)
  \reference{B}(3,3.0 )
  \separator{2.0}
  \reference{C}(2,1.0)
\end{mscdoc}
\end{verbatim}
\caption{Source code for diagrams of Figure 1}
\end{minipage}
\end{figure}

The MSC environment provides most functionality of the package. The following concepts should help in understanding the user-commands of this environment.

\textit{current height} The \textit{current height} of an MSC is a length that indicates the y-position relative to the top of the MSC frame. The MSC drawing commands use this y-position to draw MSC symbols, e.g., instance heads, messages, actions, and instance feet. The internal length \texttt{\msc@currentheight} is the current height. This variable is changed whenever the command \texttt{\nextlevel} is invoked.

\textit{current width} The \textit{current width} of an MSC is the distance from the left side of the MSC frame to the right side of the MSC frame. As such, it depends on the lengths \texttt{\envinstdist} and
\texttt{\textbackslash instdist} as well as on the number of instances. The internal length \texttt{\textbackslash msc@currentwidth} is the current width. During construction of an \texttt{\textbackslash msc}, that is, in between \texttt{\begin{msc}} and \texttt{\end{msc}}, \texttt{\textbackslash msc@currentwidth} is equal to \texttt{\textbackslash envinstdist} + \( n \times \texttt{\textbackslash instdist} \), provided that \( n \) is the number of instances defined so far (see \texttt{\textbackslash mscinstcnt} in the section \textit{Internal counters} below) and the length \texttt{\textbackslash instdist} is not changed between instances. At the end of an \texttt{\textbackslash msc} construction, an additional \texttt{\textbackslash envinstdist} is added to \texttt{\textbackslash msc@currentwidth}. The \texttt{\textbackslash msc@currentwidth} determines the x-position of new \texttt{\textbackslash msc} instances.

\textit{level} A level is a horizontal layer in an \texttt{\textbackslash msc} which is used to construct \texttt{\textbackslash msc}'s in a top-down fashion. Each level is \texttt{\textbackslash levelheight} units high and spans the complete width of the \texttt{\textbackslash msc}. The first level starts at \texttt{\textbackslash topheaddist} + \texttt{\textbackslash instheaddist} + \texttt{\textbackslash firstlevelheight} units below the top of the \texttt{\textbackslash msc} frame. The \texttt{\textbackslash nextlevel} commands advances the \texttt{\textbackslash msc} to the next (lower) level.

\textit{MSC instance} The main building blocks of MSC diagrams are \textit{instances}. Instances are represented by vertical bars. \textit{Fat} instances are represented by two vertical lines. Usually, an instance has both a head symbol and a foot symbol. In the \texttt{\textbackslash msc} macro package, each \texttt{\textbackslash msc} instance has a \textit{nickname} by which the instance is identified. In the \texttt{\textbackslash msc} macro package there are special instances:

- The left environment (nickname \texttt{envleft}).
- The right environment (nickname \texttt{envright}).
- The left side of an inline expression. If the nickname of the inline expression is \texttt{nm}, the nickname of the left side is \texttt{nmleft}.
- The right side of an inline expression. If the nickname of the inline expression is \texttt{nm}, the nickname of the left side is \texttt{nmright}.
- The left side of an reference expression. If the nickname of the reference expression is \texttt{nm}, the nickname of the left side is \texttt{nmleft}.
- The right side of an reference expression. If the nickname of the reference expression is \texttt{nm}, the nickname of the left side is \texttt{nmright}.
- A \textit{dummy} instance is an instance that is invisible; it reserves space needed to draw an instance. It is useful to create (see \texttt{create} command) instances with create-messages.

\textit{nickname} A nickname is a unique identification of an \textit{instance}.

\textit{message label reference points} In order to place a message label somewhere near the message arrow, the \texttt{\textbackslash msc} macro package computes a \textit{reference point} for each message label. This is a location on the bounding box of the label such that the distance between the arrow and the reference point is minimized. Figures 3 (page 4) and 4 (page 5) show the location of reference points for all possible locations of message labels. Note that the boxes with the location of the reference points are not generated by the \texttt{\LaTeX} code given in these figures; we enriched the \texttt{\LaTeX} code with some extra \texttt{pstricks} code (see \texttt{\LaTeX} source code of this document).

\section{Environments}

\texttt{\begin{msc}}[titlepos]{\textit{title}}...\texttt{\end{msc}} The environment to draw \texttt{\textbackslash msc}'s. The parameter \texttt{title} defines the title of the \texttt{\textbackslash msc}. The optional parameter \texttt{titlepos} defines the position of the title relative to the frame of the \texttt{\textbackslash msc}. Valid positions are \texttt{1} (left), \texttt{c} (center), and \texttt{r} (right). The default position is \texttt{1}.

\texttt{\begin{hmsc}}[titlepos]{\textit{title}}...\texttt{\end{hmsc}}(\texttt{llx, lly})(\texttt{urx, ury}) The environment to draw \texttt{\textbackslash hmsc}'s. The parameter \texttt{title} defines the title of the \texttt{\textbackslash hmsc}. The optional parameter \texttt{titlepos} defines the position of the title relative to the frame of the \texttt{\textbackslash hmsc}. Valid positions are \texttt{1} (left),
\begin{msc}
\declinst{m0}{I0}{}
\declinst{m1}{I1}{}
\declinst{m2}{I2}{}
\nextlevel
\mess{S}{m0}{m1}
\nextlevel
\mess{W}{m1}{m2}
\nextlevel[2]
\mess{S}{m1}{m0}
\nextlevel
\mess{W}{m2}{m1}
\nextlevel[2]
\mess{E}{m0}{m0}[2]
\mess{E}{m2}{m2}[2]
\nextlevel[4]
\mess{W}{m0}{m0}[2]
\mess{W}{m2}{m2}[2]
\nextlevel[6]
\mess{E}{m0}{m0}[-2]
\mess{E}{m2}{m2}[-2]
\nextlevel[4]
\mess{W}{m0}{m0}[-2]
\mess{W}{m2}{m2}[-2]
\nextlevel[6]
\mess{SW}{m0}{m1}[2]
\mess{NW}{m1}{m2}[2]
\nextlevel[6]
\mess{SW}{m1}{m0}[-2]
\mess{NW}{m2}{m1}[-2]
\nextlevel[2]
\mess{SE}{m1}{m0}[-2]
\mess{NE}{m0}{m1}[2]
\nextlevel[6]
\mess{SE}{m0}{m1}[-2]
\mess{NE}{m1}{m2}[-2]
\nextlevel[2]
\end{msc}

Figure 3: Reference points of message labels

c (center), and r (right). The default position is 1. The size of the HMSC is determined by two pairs of coordinates. The coordinates (llx,lly) define the lower left corner of the HMSC. The coordinates (urx,ury) define the upper right corner of the HMSC.

\begin{mscdoc}[titlepos][title]...\end{mscdoc}(llx,lly)(urx,ury) The environment to draw MSC documents. The parameter title defines the title of the MSCdoc document. The
optional parameter \textit{titlepos} defines the position of the title relative to the frame of the \textsc{msc} document. Valid positions are l (left), c (center), and r (right). The default position is l. The size of the \textsc{msc} document is determined by two pairs of coordinates. The coordinates $(llx,lly)$ define the lower left corner of the \textsc{msc} document. The coordinates $(urx,ury)$ define the upper right corner of the \textsc{msc} document.
4 Commands

\action(*)\{txt\}\{nm\} \quad \text{Draws an action symbol on the instance with nickname nm. The parameter txt defines the name of the action. The size of the action symbol is controlled by the \actionheight and \actionwidth lengths. The starred version expands the action symbol based on the size of the contents.}

\arrow\{nm0\}\{\{xpos1, ypos1\}...\{xposn, yposn\}\}\{nm1\} \quad \text{Draws an arrow in an HMSC diagram. The arrow starts at the symbol with nickname \textit{nm0} and ends at the symbol with nickname \textit{nm1}. The optional parameter \{xpos1, ypos1\}...\{xposn, yposn\} is a list of intermediate points the arrow should pass through.}

\changeinstbarwidth\{nm\}\{wd\} \quad \text{Changes the bar width of instance nm to \textit{wd}. The parameter \textit{wd} should be a valid \LaTeX{} length.}

\msccomment\{pos\}\{txt\}\{nm\} \quad \text{Puts a comment at instance nm. The parameter txt is the comment. The optional parameter pos defines the horizontal position of the comment relative to instance nm. Valid positions are 1 (left), r (right), and all valid lengths. If the position is 1 or r, the comment will be put at \msccommentdist units to the left or right, respectively, from the instance axis. If pos is a length, the comment will be put pos units from the instance axis. A negative pos puts the comment to the left and a positive pos puts it to the right of the instance axis.}

\condition(*)\{txt\}\{instancelist\} \quad \text{Draws a condition symbol on the instances occurring in instancelist. The parameter txt defines the text to be placed in the condition symbol. The parameter instancelist is a comma separated list of instance nicknames. Note that there should be no white space between the commas and the nicknames; only if a nickname contains white space is a white space allowed in instancelist. The starred version expands the action symbol based on the size of the contents.}

\coregionend\{nm\} \quad \text{Ends the co-region on the instance nm. This command is obsolete (see \coregionend).}

\coregionstart\{nm\} \quad \text{Starts a co-region on the instance nm. This command is obsolete (see \regionstart).}

\create\{msg\}\{labelpos\}\{creator\}\{placement\}\{nm\}\{na\}\{in\} \quad \text{Creates the instance with label \textit{nm}. If \textit{creator} is specified, a message with label \textit{msg} is sent from instance \textit{creator} to instance \textit{nm}. Instance \textit{nm} should be a dummy (invisible) instance at the time of the create message, see \dummyinst. The head symbol of \textit{nm} is drawn at \msc@currentheight. The parameter \textit{an} (above name) is put above the head symbol. The parameter \textit{in} (inside name) is put inside the head symbol. nm’s y-position is set to \msc@currentheight + \instheadheight. The optional parameter \textit{labelpos} defines the position of the message label. Valid values are t and b, denoting a label position on top of the arrow and a label position below the arrow, respectively. The optional parameter \textit{placement} defines the relative position of the message label along the message arrow. Valid values are real numbers in the closed interval [0, 1], where 0 corresponds to the beginning of the arrow and 1 corresponds to the end of the arrow. The default value is 0.5.}

\declinst(*)\{nm\}\{an\}\{in\} \quad \text{Defines an instance with nickname nm. The starred version makes a fat instance. The x-position is \instdist to the right of \msc@currentwidth. The head symbol of the instance is drawn at \msc@currentheight. The parameter \textit{an} (above name) is put above the head symbol. The parameter \textit{in} (inside name) is put inside the head symbol. The instance y-position is set to \msc@currentheight + \instheadheight.}

\drawframe\{str\} \quad \text{A command to turn on/off the drawing of the frame around msc’s, hmsc’s, and mscdoc’s. If \textit{str} is ‘yes’, the frame will be drawn, otherwise the frame will not be drawn.}
\textbf{\texttt{\textbackslash{}dravinstfoot\{str\}}} A command to turn on/off drawing of instance foot symbols. If \textit{str} is ‘yes’, the foot symbols will be drawn, otherwise they will not be drawn.

\textbf{\texttt{\textbackslash{}dravinsthead\{str\}}} A command to turn on/off drawing of instance head symbols. If \textit{str} is ‘yes’, the head symbols will be drawn, otherwise they will not be drawn.

\textbf{\texttt{\textbackslash{}dummyinst\{\textit{nm}\}}} Defines a dummy instance with nickname \textit{nm}. The starred version makes a fat instance. The \texttt{xpos} parameter is placed inside the condition symbol. The unstarred version produces an invisible connection symbol. The \texttt{txt} parameter is placed inside the condition symbol.

\textbf{\texttt{\textbackslash{}found\{pos\}\{label\}\{labelpos\}\{gate\}\{\textit{nm}\}\{placement\}}} Draws a found message to instance \textit{nm}. The \texttt{label} parameter defines the message name. The \texttt{gate} parameter defines the gate name. The optional parameter \texttt{pos} defines the position of the message relative to instance \textit{nm}. Valid positions are \texttt{l} (left) and \texttt{r} (right). The default position is \texttt{l}. The optional parameter \texttt{labelpos} defines the position of the message label with respect to the arrow. Valid values are \texttt{t} (on top) and \texttt{b} (below). The default value is \texttt{t}. The optional parameter \texttt{placement} defines the relative position of the message label along the message arrow. Valid values are real numbers in the closed interval \([0,1]\), where 0 corresponds to the beginning of the arrow and 1 corresponds to the end of the arrow. The default value is 0.5. The length of the arrow is determined by \texttt{\textbackslash{}selfmesswidth}.

\textbf{\texttt{\textbackslash{}gate\{\textit{nm}\}\{\textit{txt}\}\{xpos\}\{ypos\}}} Draws a gate at instance \textit{nm}. The \texttt{txt} parameter defines the name of the gate. The starred version produces a visible gate by drawing a black circle at instance \textit{nm}. The unstarred version produces an invisible gate. The position of the parameter \texttt{txt} is controlled by the optional parameters \texttt{hpos} and \texttt{vpos}: \texttt{hpos} defines the horizontal position relative to instance \textit{nm} and \texttt{vpos} defines the vertical position relative to the current height (\texttt{\textbackslash{}msc@currentheight}). Valid horizontal positions are \texttt{l} (left) and \texttt{r} (right). The default horizontal position is \texttt{l}. Valid vertical positions are \texttt{t} (top), \texttt{c} (center), and \texttt{b} (bottom). The default vertical is \texttt{t}.

\textbf{\texttt{\textbackslash{}hmsccondition\{\textit{nm}\}\{\textit{txt}\}\{xpos\}\{ypos\}}} Draws an HMS condition symbol with nickname \textit{nm} at position \texttt{(xpos,ypos)}. The \texttt{txt} parameter is placed inside the condition symbol.

\textbf{\texttt{\textbackslash{}hmscconnection\{\textit{nm}\}\{xpos\}\{ypos\}}} Draws an HMS connection symbol with nickname \textit{nm} at position \texttt{(xpos,ypos)}. The unstarred version produces an invisible connection symbol. The starred version produces a visible connection symbol (i.e., a small circle).

\textbf{\texttt{\textbackslash{}hmscendsymol\{\textit{nm}\}\{xpos\}\{ypos\}}} Draws an HMS end symbol with nickname \textit{nm} at position \texttt{(xpos,ypos)}.

\textbf{\texttt{\textbackslash{}hmsckeyword}} The HMS keyword. The default value is ‘hmsc’.

\textbf{\texttt{\textbackslash{}hmsckeywordstyle\{\textit{kw}\}}} A one-parameter command to typeset the HMS keyword. The command can expect \textbf{\texttt{\textbackslash{}hmsckeyword}} to be the value of \textit{kw}. The default ‘value’ is \textbf{\texttt{\textbackslash{}textbf}}.

\textbf{\texttt{\textbackslash{}hmscreference\{\textit{nm}\}\{\textit{txt}\}\{xpos\}\{ypos\}}} Draws an HMS reference symbol with nickname \textit{nm} at position \texttt{(xpos,ypos)}. The \texttt{txt} parameter is placed inside the condition symbol.

\textbf{\texttt{\textbackslash{}hmscstartsymboll\{\textit{nm}\}\{xpos\}\{ypos\}}} HMS start symbol with nickname \textit{nm} at position \texttt{(xpos,ypos)}.

\textbf{\texttt{\textbackslash{}inlineend\{\textit{nm}\}}} Ends the matching inline expression (matching means equal nicknames). The unstarred version draws a solid line to close the inline expression. The starred version draws a dashed line to close the inline expression.

\textbf{\texttt{\textbackslash{}inlineseparator\{\textit{nm}\}}} Draws an inline separator line at the inline expression with nickname \textit{nm}. The separator is drawn at \texttt{\textbackslash{}msc@currentheight}.  

7
\inlinestart{lo}{ro}{nm}{\text{txt}}{fi}{li}  Defines an inline expression with nickname \textit{nm}. The inline expression is started at $\texttt{\textbackslash msc@currentheight}$ and continues until the level where a matching \texttt{\inlinestop} command is found (matching means equal nicknames). The \textit{txt} parameter defines the text of the inline expression. The first instance of the inline expression is \textit{fi}. The last instance of the inline expression is \textit{li}. The optional parameter \textit{lo} defines the left and right overlap of the inline expression. If the second optional parameter, \textit{ro}, is present, \textit{lo} defines the left and \textit{ro} defines the right overlap.

\inststart{nm}{an}{\text{in}}  Starts instance with nickname \textit{nm}. Instance \textit{nm} should be a dummy (invisible) instance at the time of the \texttt{\inststart} command, see \texttt{\dummyinst}. The head symbol is drawn at $\texttt{\textbackslash msc@currentheight}$. The parameter \textit{an} (above name) is put above the head symbol. The parameter \textit{in} (inside name) is put inside the head symbol. The instance y-position is set to $\texttt{\textbackslash msc@currentheight + \instheadheight}$.

\inststop{nm}  Stops instance with nickname \textit{nm}. The foot symbol is drawn at $\texttt{\textbackslash msc@currentheight}$. The instance y-position is undefined after this command.

\lost{pos}{label}{\text{labpos}}{\text{gate}}{\text{nm}}{\text{placement}}  Draws a lost message from instance \textit{nm}. The \textit{label} parameter defines the message name. The \textit{gate} parameter defines the gate name. The optional parameter \textit{pos} defines the position of the message relative to instance \textit{nm}. Valid positions are \texttt{l} (left) and \texttt{r} (right). The default position is \texttt{t}. The optional parameter \textit{labpos} defines the position of the message label with respect to the arrow. Valid values are \texttt{t} (on top) and \texttt{b} (below). The default value is \texttt{t}. The optional parameter \textit{placement} defines the relative position of the message label along the message arrow. Valid values are real numbers in the closed interval $[0,1]$, where 0 corresponds to the beginning of the arrow and 1 corresponds to the end of the arrow. The default value is 0.5. The length of the arrow is determined by $\texttt{\textbackslash selfmesswidth}$.

\measure{\star}{pos}{\text{txt}}{\text{nm1}}{\text{nm2}}{\text{offset}}  Puts a measure at instances \textit{nm1} and \textit{nm2}. The parameter \textit{txt} defines the label of the measure. The starred version puts the triangular measure symbols outside the measure; the unstared version puts the triangular measure symbols inside the measure. The optional \textit{pos} parameter defines the horizontal position of the measure relative to instances \textit{nm1} and \textit{nm2}. Valid positions are \texttt{l} (left), \texttt{r} (right), and all valid lengths. If the position is \texttt{l} or \texttt{r}, the measure will be put at $\texttt{\textbackslash measuredist}$ units to the left or right, respectively, from the closest instance axis. If \textit{pos} is a length, the measure will be put \textit{pos} units from the closest instance axis. A negative \textit{pos} puts the measure to the left and a positive \textit{pos} puts it to the right of the instances. The optional parameter \textit{offset} defines the number of levels the measure should extend vertically. The default value for \textit{offset} is 1. Valid values for \textit{offset} are all real numbers.

\measureend{\star}{pos}{\text{txt}}{\text{nm}}{\text{gate}}  Puts a measure end symbol at instance \textit{nm}. The starred version puts the triangular measure symbol outside the measure; the unstared version puts the triangular measure symbol inside the measure. The \textit{txt} parameter defines the label of the measure. The \textit{gate} parameter defines the name of the gate of the measure end symbol. The optional \textit{pos} parameter defines the horizontal position of the measure relative to the \textit{nm} instance. Valid positions are \texttt{l} (left), \texttt{r} (right), and all valid lengths. If the position is \texttt{l} or \texttt{r}, the measure will be put at $\texttt{\textbackslash measuredist}$ units to the left or right, respectively, from the instance axis. If \textit{pos} is a length, the measure will be put \textit{pos} units from the instance axis. A negative \textit{pos} puts the measure to the left and a positive \textit{pos} puts it to the right of the instance.

\measuresstart{\star}{pos}{\text{txt}}{\text{nm}}{\text{gate}}  Puts a measure start symbol at instance \textit{nm}. The starred version puts the triangular measure symbol outside the measure; the unstared version puts the triangular measure symbol inside the measure. The \textit{txt} parameter defines the label of the measure. The \textit{gate} parameter defines the name of the gate of the measure start symbol. The optional parameter \textit{pos} defines the horizontal position of the measure relative to instance \textit{nm}. Valid positions are \texttt{l} (left), \texttt{r} (right), and all valid lengths. If the position is \texttt{l} or \texttt{r}, the
\textbf{mscmark} A one-parameter command to typeset the \texttt{mscmark} keyword. The command can expect \texttt{\mscmark} to be the value of \texttt{kw}. The default ‘value’ is \texttt{textbf}.

\textbf{mscdate} The date of the \texttt{MSC} macro package.

\textbf{mscdockeyword} The \texttt{MSCdoc} keyword. The default value is ‘\texttt{mscdoc}’.

\textbf{mscdockeywordstyle} A one-parameter command to typeset the \texttt{mscdoc} keyword. The command can expect \texttt{\mscdockeyword} to be the value of \texttt{kw}. The default ‘value’ is \texttt{textbf}.

\textbf{msckeyword} The \texttt{MSC} keyword. The default value is ‘\texttt{msc}’.

\textbf{msckeywordstyle} A one-parameter command to typeset the \texttt{MSC} keyword. The command can expect \texttt{\msckeyword} to be the value of \texttt{kw}. The default ‘value’ is \texttt{textbf}.

\textbf{mscmark} Puts a mark at instance \texttt{nm}. The parameter \texttt{txt} is the name of the mark. The optional parameter \texttt{pos} defines the horizontal and vertical position of the mark relative to instance \texttt{nm} and the current height \texttt{\msc\@currentheight}. Valid positions are \texttt{t1} (top-left), \texttt{tr} (top-right), \texttt{bl} (bottom-left), and \texttt{br} (bottom-right). The default position is \texttt{t1}. The horizontal distance between the mark and the instance is defined by \texttt{\markdist}.

\textbf{mscunit} A string denoting the (default) unit of all lengths used by the \texttt{MSC} macro package. Valid values are \texttt{cm}, \texttt{em}, \texttt{ex}, \texttt{in}, \texttt{mm}, \texttt{pt}, etc. The default value is \texttt{cm}.

\textbf{action} Draws an \texttt{action} symbol with a cross on the instance with nickname \texttt{nm}. The parameter \texttt{txt} defines the name of the action. The size of the action symbol is controlled by the \texttt{\actionheight} and \texttt{\actionwidth} lengths. The starred version expands the action symbol based on the size of the contents.

\textbf{condition} Draws a \texttt{condition} symbol with a cross on the instances occurring in \texttt{instancelist}. The parameter \texttt{txt} defines the text to be placed in the \texttt{condition} symbol. The parameter \texttt{instancelist} is a comma separated list of instance nicknames. Note that there should be no white space between the commas and the nicknames; only if a nickname contains white space is a white space allowed in \texttt{instancelist}. The starred version expands the action symbol based on the size of the contents.

\textbf{measuredist} The measure will be put at \texttt{\measuredist} units to the left or right, respectively, from the instance axis. If \texttt{pos} is a length, the measure will be put \texttt{pos} units from the instance axis. A negative \texttt{pos} puts the measure to the left and a positive \texttt{pos} puts it to the right of the instance.

\texttt{\mess(\ast)[pos][\{label\}][labelpos][\{sender\}][placement][\{receiver\}][\offset]} Draws a message from \texttt{\sender} instance to \texttt{\receiver} instance. The starred version draws a dashed line arrow, instead of a solid arrow. This can be used to distinguish method calls from method replies. The \texttt{\sender} and \texttt{\receiver} may be the same instance, in which case the message is a self message. The parameter \texttt{\label} defines the message name. The message starting y-position is \texttt{\msc\@\currentheight} and the ending y-position of the message is defined by \texttt{\msc\@\currentheight + (\offset \times \levelheight)}. The optional parameter \texttt{pos} defines the position of self messages with respect to the instance axis. Valid values are \texttt{l} (left) and \texttt{r} (right). The default value is \texttt{l}. The optional parameter \texttt{labelpos} defines the position of the message label. In case of a self message, valid values are \texttt{l} and \texttt{r}, denoting a label position left from the arrow and right from the arrow, respectively. For self-messages the default value of \texttt{\labelpos} is the value of \texttt{\pos}. In case of a non-self message, valid values are \texttt{t} (default) and \texttt{b}, denoting a label position on top of the message arrow and below the message arrow, respectively. The optional parameter \texttt{placement} defines the relative position of the message label along the message arrow. Valid values are real numbers in the closed interval \([0,1]\), where \texttt{0} corresponds to the beginning of the arrow and \texttt{1} corresponds to the end of the arrow. The default value is \texttt{0.5}. The default value of the optional parameter \texttt{\offset} is \texttt{0} for normal messages and \texttt{1} for self messages. Valid values for \texttt{\offset} are all real numbers.

\textbf{messarrowscale} Sets the scale factor (a real number) of message arrow heads. The default value is \texttt{1.5}.
\setmscunit{unit} Changes the value of \mscunit into unit. Valid values for unit are cm, em, ex, in, mm, pt, etc.

\mscversion The version number of the MSC macro package.

\condition{(*)}{txt}{instance} Draws a crossed-out condition. Its functionality is otherwise equal to the condition command.

\nextlevel{offset} Increases the number of levels by the value of the optional parameter offset. Valid values for offset are all real numbers. The default value of offset is 1. Increasing the level number means that \currentheight is increased by offset × \levelheight. The first time this macro is used, the actual increase of \currentheight is firstlevelheight + (offset − 1) × \levelheight). Negative values of offset back up a number of levels. There are situations where this is useful, see Section 10.

\nogrid Turns off grid drawing in MSC, HMSC, and MSCdoc diagrams. This command should not be used inside an MSC, HMSC, or MSCdoc environment.

\order{pos}{sender}{receiver}{offset} Draws an order line from the sender instance to the receiver instance. The sender and receiver may be the same instance, in which case the order is a self-order. The order starting y-position is \currentheight and the ending y-position of the order is defined by \currentheight + (offset × \levelheight). In case of a self-order, the optional parameter pos defines the position of the order relative to the sender instance. Valid positions are 1 (left) and r (right). The default position is 1. In case of a non-self-order, the pos parameter is ignored. The default value of the optional parameter offset is 0 for normal orders and 1 for self orders. Valid values for offset are all real numbers.

\reference{txt}{xpos,ypos} Draws an MSCdoc reference symbol. The txt parameter defines the text to be placed inside the MSCdoc reference symbol. The coordinates (xpos,ypos) define the position of the reference symbol.

\referred{nm} Ends the reference expression with nickname nm.

\referredstart[lo][ro]{nm}{txt}{fi}{li} Defines a reference expression with nickname nm. The reference expression is started at \currentheight and continues until the level where a matching \referredend command is found. The txt parameter defines the text of the reference expression. The first instance of the reference expression is fi. The last instance of the reference expression is li. The optional parameter lo defines the left and right overlap of the reference expression. If the second optional parameter, ro, is present, lo defines the left and ro defines the right overlap.

\regionend{nm} Ends the current region on instance nm. The region style of the instance nm is reset to normal again. Note: this command makes \coregionend obsolete.

\regionstart{rstyle}{nm} Starts a region on the instance nm. The style of the region is defined by the rstyle parameter. Valid region styles are coregion, suspension, activation, and normal. Note: this command makes \coregionstart obsolete.

\separator{ypos} Draws a separator in an MSCdoc diagram. The ypos parameter defines the vertical position of the separator in the MSCdoc diagram.

\setfootcolor{color} Sets the color of the foot symbols of MSC instances. Possible values are black, white, gray, or lightgray. For more color values, see the documentation of the LATEX \footcolor package.

\setmsckeyword{kw} Sets the HMSC keyword to kw. For this command to be effective, it should be used outside the HMSC environment.
\sethmsc{kwstylemacro} Redefines the \hmsc{kwstylemacro} macro to the macro \kwstylemacro. This should be a 1-argument macro, like the standard \LaTeX \textbf{textbf} and \textit commands. For this command to be effective, it should be used outside the HMSC environment.

\setmsc{kw} Sets the MScdoc keyword to \kw. For this command to be effective, it should be used outside the MScdoc environment.

\setmsc{kwstylemacro} Redefines the \msc{kwstylemacro} macro to the macro \kwstylemacro. This should be a 1-argument macro, like the standard \LaTeX \textbf{textbf} and \textit commands. For this command to be effective, it should be used outside the MScdoc environment.

\setmsc{kw} Sets the MSc keyword to \kw. For this command to be effective, it should be used outside the MSc environment.

\setmsc{kwstylemacro} Redefines the \msc{kwstylemacro} macro to the macro \kwstylemacro. This should be a 1-argument macro, like the standard \LaTeX \textbf{textbf} and \textit commands. For this command to be effective, it should be used outside the MSc environment.

\setmsc{scale}{scalefactor} Sets the scale factor of the MSc environment to \scalefactor. The scale factor is supposed to be a real number. Scaling is done when the MSc environment ends (\end{msc}). The default of \scalefactor is 1.

\setmsc{values}{size} Sets the msc-lengths to one of the predefined \sizes. Valid values for \size are: \textbf{small}, \textbf{normal}, and \textbf{large}.

\setstop{pos}{label}{nm}{offset} Draws both a \textbf{timer} and a \textbf{stop timer} symbol on the instance \textit{nm}. The parameter \textit{label} defines the name of the timer. The optional parameter \textit{pos} defines the position of the \textbf{timer} relative to the \textit{nm} instance. Valid positions are \textbf{l} (left) and \textbf{r} (right). The default position is \textbf{l}. The horizontal distance between the timer symbol and the instance axis is defined by \textit{selfmesswidth}. The optional parameter \textit{offset} defines the number of levels between the \textbf{timer} symbol and the point where the arrow meets the \textit{nm} instance. Valid values for \textit{offset} are all real numbers. The default \textit{offset} is 2.

\settimeout{pos}{label}{nm}{offset} Draws a \textbf{timer} symbol on the instance \textit{nm} and connects the \textbf{timer} symbol and the instance with an arrow. The parameter \textit{label} defines the name of the \textbf{timer}. The optional parameter \textit{pos} defines the position of the \textbf{timer} relative to the \textit{nm} instance. Valid positions are \textbf{l} (left) and \textbf{r} (right). The default position is \textbf{1}. The optional parameter \textit{offset} defines the number of levels between the \textbf{timer} symbol and the point where the arrow meets the \textit{nm} instance. Valid values for \textit{offset} are all real numbers. The default \textit{offset} is 2. The horizontal distance between the timer symbol and the instance axis is defined by \textit{selfmesswidth}.

\settimer{pos}{label}{nm} Draws a \textbf{timer} symbol on the instance \textit{nm}. The parameter \textit{label} defines the name of the \textbf{timer}. The optional parameter \textit{pos} defines the position of the \textbf{timer} relative to the \textit{nm} instance. Valid positions are \textbf{l} (left) and \textbf{r} (right). The default position is \textbf{1}. The horizontal distance between the timer symbol and the instance axis is defined by \textit{selfmesswidth}.

\showgrid Turns on grid-drawing in MSc, MScdoc, and HMSC diagrams. This is useful to determine the values of the user definable lengths or if normal \pstricks commands should be included in the diagram. (Note that the vertical axis of the MSc grid has no positive labels.) This command should not be used inside an MSc, HMSC, or MScdoc environment.

\stop\textbf{(*)}\textit{nm} Stops the instance with nickname \textit{nm}. The instance line of \textit{nm} is drawn from its \textit{y}-position to the current \textit{y}-position of the MSc (\textbf{\msc@currentheight}). At the current
height, a \textit{stop} symbol is drawn. The starred version draws an \textit{instance foot} symbol instead of a \textit{stop} symbol.

\texttt{\textbackslash stop\texttt{timer}}[\texttt{pos}]{\texttt{\{label\}}}\texttt{nm} \quad \text{Draws a \textit{stop timer} symbol on the instance \texttt{nm}. The parameter \texttt{label} defines the name of the timer. The optional parameter \texttt{pos} defines the position of the \textit{timer} relative to the \texttt{nm} instance. Valid positions are \texttt{l} (left) and \texttt{r} (right). The default position is \texttt{l}. The horizontal distance between the timer symbol and the instance axis is defined by selfmesswidth.}

\texttt{\textbackslash timeout}[\texttt{pos}]{\texttt{\{label\}}}\texttt{nm} \quad \text{Draws a \textit{timer} symbol on the instance \texttt{nm} and connects the symbol and the instance with an arrow. The parameter \texttt{label} defines the name of the timeout. The optional parameter \texttt{pos} defines the position of the \textit{timer} symbol relative to the \texttt{nm} instance. Valid positions are \texttt{l} (left) and \texttt{r} (right). The default position is \texttt{l}. The horizontal distance between the timer symbol and the instance axis is defined by selfmesswidth.}

\section{User definable lengths}

This section lists the user-definable lengths of the msc macro package. For each length, the default values for large, normal, and small diagrams are given. The appearance of msc, hmsc, and mscdoc diagrams can be changed by adjusting these lengths. Use the normal \texttt{\setlength} command to change these lengths.

\texttt{\textbackslash actionheight} \quad \text{Height of action symbols.} \\
\texttt{(large/normal/small value 0.75/0.6/0.5 cm.)}

\texttt{\textbackslash actionwidth} \quad \text{Width of action symbol.} \\
\texttt{(large/normal/small value 1.25/1.25/1.2 cm.)}

\texttt{\textbackslash bottomfootdist} \quad \text{Distance between bottom of foot symbol and frame.} \\
\texttt{(large/normal/small value 1.0/0.7/0.5 cm.)}

\texttt{\textbackslash commentdist} \quad \text{Distance between a comment and its instance.} \\
\texttt{(large/normal/small value 0.5/0.5/0.5 cm.)}

\texttt{\textbackslash conditionheight} \quad \text{Height of condition symbols.} \\
\texttt{(large/normal/small value 0.75/0.6/0.5 cm.)}

\texttt{\textbackslash conditionoverlap} \quad \text{Overlap of condition symbol.} \\
\texttt{(large/normal/small value 0.6/0.5/0.4 cm.)}

\texttt{\textbackslash envinstdist} \quad \text{Distance between environments and nearest instance line.} \\
\texttt{(large/normal/small value 2.5/2.0/1.2 cm.)}

\texttt{\textbackslash firstlevelheight} \quad \text{Height of level just below head symbols. Should not be changed inside the msc environment.} \\
\texttt{(large/normal/small value 0.75/0.6/0.4 cm.)}

\texttt{\textbackslash gatesymbolradius} \quad \text{Radius of the gate symbol.} \\
\texttt{(large/normal/small value 0.5/0.5/0.5 mm.)}

\texttt{\textbackslash hmsscconditionheight} \quad \text{Height of HMSC condition symbol.} \\
\texttt{(large/normal/small value 0.375/0.3/0.25 cm.)}

\texttt{\textbackslash hmsscconditionwidth} \quad \text{Width of HMSC condition symbol.} \\
\texttt{(large/normal/small value 1.0/0.8/0.7 cm.)}

\texttt{\textbackslash hmsscconnectionradius} \quad \text{Radius of HMSC connection symbol.} \\
\texttt{(large/normal/small value 0.06/0.05/0.04 cm.)}
\hmscreferenceheight  Height of HMSC reference symbol.
  (large/normal/small value 0.8/0.7/0.6 cm.)

\hmscreferencewidth  Width of HMSC reference symbol.
  (large/normal/small value 1.6/1.4/1.2 cm.)

\hmscstartsymbolwidth  Width of HMSC start symbol.
  (large/normal/small value 0.85/0.7/0.4 cm.)

\inlineoverlap  Overlap of inline symbol.
  (large/normal/small value 1.5/1.0/0.75 cm.)

\instbarwidth  Default width of vertical instance bars (applies to fat instances only).
  (large/normal/small value 0.0/0.0/0.0 cm.)

\instdist  Distance between instance axes.
  (large/normal/small value 3.0/2.2/1.5 cm.)

\instfootheight  Height of foot symbols. Should not be changed inside the MSC environment.
  (large/normal/small value 0.25/0.2/0.15 cm.)

\instheadheight  Height of head symbols. Should not be changed inside the MSC environment.
  (large/normal/small value 0.6/0.55/0.5 cm.)

\instwidth  Width of header and foot symbols.
  (large/normal/small value 1.75/1.6/1.2 cm.)

\labeldist  Distance between labels and the symbols to which they belong (for instance, message labels and arrows).
  (large/normal/small value 1.0/1.0/1.0 ex.)

\lastlevelheight  Height of level just above foot symbols. Should not be changed inside the MSC environment.
  (large/normal/small value 0.5/0.4/0.3 cm.)

\leftnamedist  Distance between left of the frame and (left of) MSC, HMSC, or MSCdoc title.
  (large/normal/small value 0.3/0.2/0.1 cm.)

\levelheight  Height of a level.
  (large/normal/small value 0.75/0.5/0.4 cm.)

\lostsymbolradius  Radius of the lost and found symbols.
  (large/normal/small value 0.15/0.12/0.08 cm.)

\markdist  Horizontal distance from a mark to its instance.
  (large/normal/small value 1.0/1.0/1.0 cm.)

\measuredist  Horizontal distance from a measure to its (closest) instance.
  (large/normal/small value 1.0/1.0/1.0 cm.)

\measuresymbolwidth  Width of a measure symbol.
  (large/normal/small value 0.75/0.6/0.4 cm.)

\mscdocreferenceheight  Height of reference symbol in an MSCdoc.
  (large/normal/small value 0.8/0.7/0.6 cm.)

\mscdocreferencewidth  Width of reference symbol in an MSCdoc.
  (large/normal/small value 1.6/1.4/1.2 cm.)
\texttt{\textbackslash referenceoverlap} Overlap of reference symbol.
\hspace{1cm} (large/normal/small value 1.5/1.0/0.75 cm.)

\texttt{\textbackslash regionbarwidth} Width of region bars.
\hspace{1cm} (large/normal/small value 0.5/0.4/0.2 cm.)

\texttt{\textbackslash selfmesswidth} Length of horizontal arms of self-messages, self-orders, lost messages and found messages as well as horizontal distance between instance axis and timer symbols.
\hspace{1cm} (large/normal/small value 0.75/0.6/0.4 cm.)

\texttt{\textbackslash stopwidth} Width of the stop symbol.
\hspace{1cm} (large/normal/small value 0.6/0.5/0.3 cm.)

\texttt{\textbackslash timerwidth} Width of the \textit{timer} symbols.
\hspace{1cm} (large/normal/small value 0.4/0.3/0.2 cm.)

\texttt{\textbackslash topheaddist} Distance between top of head symbols and frame.
\hspace{1cm} (large/normal/small value 1.5/1.3/1.2 cm.)

\texttt{\textbackslash toprnamedist} Distance between top of the frame and (top of) MSC, HMSC, or MSC doc title.
\hspace{1cm} (large/normal/small value 0.3/0.2/0.2 cm.)

\section{Internal lengths}

The \texttt{MSC} macro package uses some scratch lengths to perform calculations. Below, these scratch lengths are listed.

\texttt{\textbackslash msc@commentdist} Internal length to compute distance between comments and instances.
\hspace{1cm} (This length should be removed in the future.)

\texttt{\textbackslash msc@currentheight} The current height of the current MSC environment.

\texttt{\textbackslash msc@currentwidth} The current width of the current MSC environment.

\texttt{\textbackslash msc@totalheight} The final height of the current MSC environment.

\texttt{\textbackslash msc@totalwidth} The final width of the current MSC environment.

\texttt{\textbackslash tmp@X} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Xa} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Xb} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Xc} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Xd} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Y} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Ya} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Yb} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Yc} Scratch length for intermediate computations.

\texttt{\textbackslash tmp@Yd} Scratch length for intermediate computations.
7 Internal boxes

\mscbox  The box that contains the current MSC just before it is put on paper.
\tmp@box  Scratch box for intermediate computations

8 Internal counters

\mscinstcnt  The MSC instance counter. This counter is increased each time an instance is created.
\tmpcnt  Scratch counter for intermediate computations.

9 Limitations

1. The frames in an MSC do not automatically scale with the text inside the frame. However, the size of the frames can be set manually.

2. Start and end points of messages are computed at the current level. This can give ill-looking effects if the width of the bar of an instance changes after the message is drawn, e.g., if an activation region starts or ends after the message is drawn.

3. Messages that cause the start of a region should be drawn after the \regionstart command, but in the same level.

4. Messages that denote the end of a region should be drawn before the \regionend command.

5. Activation regions make crossing messages partly invisible. A solution for this problem is to first draw the instance foot symbols at the right level (using \inststop{i}), then back up the total number of levels of the MSC (using \nextlevel{-n}), and then drawing the messages.

6. Documents using the MSC macro package cannot be compiled with pdflatex. The reason for this is that pstricks is not supported by pdflatex.

7. The source code of the MSC macro package is only marginally documented. Therefore, changes/improvements by others are unlikely.

10 Tricks

In this section we describe some tricks to use the MSC macro package efficiently.

Multi-line text arguments  Many graphical objects in MSC diagrams have text labels. In general, the commands to draw these objects put the text arguments on one line. If the text should consist of multiple lines, the \LaTeX \parbox command can be used. For instance, to generate a message with a two-line label, write:

\mess{\parbox{1cm}{two\ lines}}{s}{r}

The following \LaTeX code snippet defines the command \resizebox. The resize box defines a \parbox and computes its own width parameter.
Specifying lengths  The msc macro package imports the calc package in order to have a more natural syntax for arithmetical expressions. Consequently, if a command expects a \TeX length argument, it is possible to use the expression syntax offered by calc.

For example, consider the msc of Figure 5. To make sure the comment for instance $j$ appears 1ex to the right of the msc frame, the value of the optional pos parameter of the \comment command should be

\instdist + \envinstdist + 1ex.

To express this in normal \TeX, one should write something like

\newlength{\l}
\setlength{\l}{\instdist}
\addtolength{\l}{\envinstdist}
\addtolength{\l}{1ex}
\msccomment{\l}{Comment for $j$}{j}

inside the msc code. However, using calc’s expression syntax, it is also possible to write

\msccomment{\instdist + \envinstdist + 1ex}{Comment for $j$}{j}

The complete code for the diagram of Figure 5 is given below. Since the calc package is included in the standard \TeX distribution, there should be no compatibility problems.
Level backup It is possible to back-up several levels: just use a negative value in the `\nextlevel` command. This feature can be useful to draw messages over regions instead of regions over messages. Compare the diagrams of Figure 6. The code for these diagrams is given below.

\begin{msc} (Specifying lengths)
\declinst{i}{}{i}
\declinst{j}{}{j}
\declinst{k}{}{k}
\nextlevel
\msccomment{instdist + envinstdist + 1ex}{Comment for $j$}
\nextlevel[2]
\end{msc}

\begin{msc} (Invisible message label)
\declinst{i}{}{i}
\declinst{j}{}{j}
\declinst{k}{}{k}
\regionstart{activation}{j}
\nextlevel
\mess{Message a}{i}[0.25]{k}[2]
\nextlevel[2]
\regionend{j}
\nextlevel
\end{msc}

\begin{msc} (Level backup makes it visible)
\declinst{i}{}{i}
\declinst{j}{}{j}
\declinst{k}{}{k}
\regionstart{activation}{j}
\nextlevel[3]
\regionend{j}
\nextlevel[-2]% backing up
\mess{Message a}{i}[0.25]{k}[2]
\nextlevel[2]% fast forward
\nextlevel
\end{msc}

\begin{msc} (Level backup makes it visible)
\declinst{i}{}{i}
\declinst{j}{}{j}
\declinst{k}{}{k}
\regionstart{activation}{j}
\nextlevel[3]
\regionend{j}
\nextlevel[-2]% backing up
\mess{Message a}{i}[0.25]{k}[2]
\nextlevel[2]% fast forward
\nextlevel
\end{msc}

Figure 6: Level back-up
References
