KEY CENTRE FOR SOFTWARE TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF QUEENSLAND

St. Lucia
Queensland
Australia 4067

TECHNICAL REPORT

No. 167

Specifying Physical Limitations
A case study of an oscilloscope

Ian Hayes

July, 1990 (revised 1993)

Specifying Physical Limitations: A Case Study of an Oscilloscope

Ian Hayes*

Abstract

This paper extends the work on developing specifications of electronic instruments of Norman Delisle and David Garlan from Tektronix. As with their original, the specification abstracts away from user interface and implementation issues, and the specification is simplified by considering the behaviour of the oscilloscope over all time, rather than incrementally.

We present a specification of a simple, single-channel, single-trigger oscilloscope in two phases:

1. the specification of an ideal oscilloscope — this makes use of concepts from real analysis to capture the ideal behaviour of an oscilloscope, and
2. the relaxation of the conditions of the specification to an extent that allows implementation within the bounds of physical limitations.

The main contribution of our paper is the treatment of physical limitations in the second phase; these turn an unimplementable ideal oscilloscope into a realistic oscilloscope that can be physically implemented.

As with the work of Delisle and Garlan the Z specification notation is used. A notable stylistic difference, however, is that, while Delisle and Garlan used functions as their specification building blocks, here we use Z schemas. Although this is mainly a stylistic difference, it does allow us to treat both deterministic and non-deterministic components of the specification in a similar fashion.

 The notation in this 1993 printing has been revised to be closer to the Z standard.

In [DG90, DG88] Norman Delisle and David Garlan from Tektronix develop specifications of ideal electronic instruments using the Z specification notation [Spi92, Hay93]. For the first phase of the specification presented in Section 1 we follow the structure developed in [DG90] of an ideal oscilloscope. This specification is relaxed in Section 2 to allow for implementation within the bounds of physical limitations.

1 An ideal oscilloscope

For our ideal oscilloscope we assume perfect resolution both in terms of time and signal levels; we model signal voltage by real numbers ($\mathbb{R}$) and time by non-negative, real numbers ($\mathbb{R}_+$). A summary of the Z notation used is given in Appendix A.

$$
\begin{aligned}
&\mathit{Voltage} == \mathbb{R} \times \{\mathit{Volt}\} \\
&\mathit{Time} == \mathbb{R}_+ \times \{\mathit{Second}\}
\end{aligned}
$$

As we are describing a physical system it is useful to include physical units; the notation used here for physical units is described in Appendix B. We assume that all the standard arithmetic operators and relations are extended to work on quantities with compatible units.

In Section 1.1 we develop a specification of how a single trace is ideally displayed. Section 1.2 covers the use of a trigger input to select the segments of the input signal to be displayed.

1.1 Displaying a single trace

An oscilloscope displays a trace by mapping time to a horizontal distance across the screen and voltage to a vertical offset on the screen. The transformations applied to a signal to derive a trace are

scaling both horizontal (seconds per metre) and vertical (volts per metre) scaling are used to convert a voltage versus time signal to a point-on-screen versus time display of the trace;

translation of the trace on the display by horizontal and vertical offsets; and

clipping the trace to fit on the screen.

In the remainder of this section we develop specifications for these transformations. We start with our representations of signals and traces.

1.1.1 Signals, segments, coordinates and traces

An input signal can be modelled as a function of time; we define the type *Signal* by

$$Signal == Time \mathrel{+\!\!\!\!\rightarrow} Voltage$$

A segment to be displayed corresponds to a contiguous time interval selected from a signal. We first define a non-empty, contiguous time period and then a contiguous segment of a signal.

$$\begin{aligned}&ContigPeriod == \{st : \mathbb{P}_1\ Time \mid \\ &\qquad (\exists\, t1, t2 : Time \bullet st = \{t : Time \mid t1 < t < t2\})\}\\ &Segment == \{s : Signal \mid \operatorname{dom} s \in ContigPeriod\}\end{aligned}$$

A coordinate can be represented by a real

$$Coord == \mathbb{R} \times \{Metre\}$$

and a point on the screen by a pair of coordinates.

$$Point == Coord \times Coord$$

A trace is a mapping from time to points.

$$Trace == Time \mathrel{+\!\!\!\!\rightarrow} Point$$

1.1.2 Scaling

Scaling takes a contiguous segment of a signal and scales it both horizontally and vertically. It also adjusts the trace so that the start of the segment corresponds to a horizontal offset of zero. The horizontal scale factor converts the units from seconds to metres; a typical value might be 5 nanoseconds per centimetre. Similarly, the vertical scale converts from a voltage to metres; a typical value might be 2 volts per centimetre.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Scale}} \\
\hline
segment : Segment \\
HScale : \mathbb{R}_{+} \times \{Second \mathbin{/} Metre\} \\
VScale : \mathbb{R} \times \{Volt \mathbin{/} Metre\} \\
scaled : Trace \\
\hline
scaled = (\lambda\, t : \operatorname{dom} segment \bullet \\
\qquad (t - inf(\operatorname{dom} segment) \mathbin{/} HScale, segment(t) \mathbin{/} VScale)) \\
\hline
\end{array}
$$

In Z, the above box is referred to as a schema: it collects together a set of typed variables together with a constraint relating the variables. We use the variables here to represent observations of the behaviour of the oscilloscope. The function *inf* gives the greatest lower bound of a non-empty, contiguous time period (see Appendix A).

1.1.3 Translation

Translation by horizontal and vertical offsets is straightforward.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Translate}} \\
\hline
scaled : Trace \\
HOffset, VOffset : Coord \\
moved : Trace \\
\hline
moved = (\lambda\, t : \operatorname{dom} scaled \bullet \\
\qquad (first(scaled(t)) + HOffset, second(scaled(t)) + VOffset)) \\
\hline
\end{array}
$$

1.1.4 Clipping

The finite size of the oscilloscope screen requires clipping of the trace to fit on the screen. We assume the horizontal axis goes from zero to HMax, while the vertical axis is symmetric about a centre of zero. For our example, we assume the following screen size definitions.

$$
\begin{array}{l}
HMax == (0.2, Metre) \\
VMax == (0.1, Metre)
\end{array}
$$

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Clip}} \\
\hline
moved : Trace \\
clipped : Trace \\
\hline
\textbf{let } screen == \\
\quad \{x, y : Coord \mid 0 < x < HMax \land -VMax < y < VMax\} \bullet \\
\qquad clipped = moved \rhd screen \\
\hline
\end{array}
$$

The operator "$\rhd$" is range restriction: the function *moved* is restricted so that the range of the resulting function is contained in the set given as the second argument. That is, only pairs in *moved* whose second component is in the set are included in the result.

1.1.5 Trace display

The collected transformations of a trace are given by the following.

$$DisplayTrace \mathrel{\widehat{=}} (Scale \land Translate \land Clip) \setminus (scaled, moved)$$

The three schemas *Scale*, *Translate* and *Clip* are combined using schema conjunction ($\land$). The result is a schema which includes the merged declarations of all three schemas: variables with the same name must have identical types; and the predicates of the three schemas are conjoined to give the combined constraint. The *scaled* and *moved* traces are hidden using schema variable hiding ($\setminus$) because they are only used as intermediate links in defining the final trace. An expanded form of *DisplayTrace* is given in Appendix A.

For future reference we gather all the parameters used for determining the transformation applied to the trace into a single schema.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{DisplayKnobs}} \\
\hline
HScale : \mathbb{R}_{+} \times \{Second \,/\, Metre\} \\
VScale : \mathbb{R} \times \{Volt \,/\, Metre\} \\
HOffset, VOffset : Coord \\
\hline
\end{array}
$$

1.2 Selecting display segments

An oscilloscope displays only a sequence of non-overlapping segments of the input signal. Each of these segments is relative to a trigger point on an input trigger signal. For our simple oscilloscope a trigger point is detected when the trigger signal passes a given threshold level with a given slope.

The following is a realistic assumption that may be made about the input signal. Amongst other things, it guarantees that the signal is differentiable, a necessary assumption for our definition of trigger point selection below.

$$
\begin{array}{l}
\mathit{DiffSignal} == \{ s : \mathit{Segment} \mid \\
\quad (\exists B : \mathbb{R} \bullet (\forall t1, t2 : \operatorname{dom} s \bullet \\
\qquad t1 \neq t2 \Rightarrow \operatorname{abs}((s(t1) - s(t2)) \,/\, (t1 - t2)) < B))\}
\end{array}
$$

1.2.1 Detecting trigger points

We can now select out the set of all possible trigger points: times at which the signal crosses a given level with a given slope. The function *deriv trigger* is the derivative of *trigger* with respect to time.

$$
\begin{array}{|l}
\hline
\textit{DetectTrig} \\
\mathit{trigger} : \mathit{DiffSignal} \\
\mathit{Level} : \mathit{Voltage} \\
\mathit{Slope} : \{-1, 1\} \\
\mathit{triggerpoints} : \mathbb{P}\, \mathit{Time} \\
\hline
\mathit{triggerpoints} = \\
\quad \{ t : \operatorname{dom} \mathit{trigger} \mid \\
\qquad \mathit{trigger}(t) = \mathit{Level} \wedge (\mathit{deriv\,trigger})(t) * \mathit{Slope} \geq 0 \} \\
\hline
\end{array}
$$

Not all trigger points need be used in determining the sequence of segments to be displayed. For example, there may be multiple trigger points in a single segment; only the initial trigger point for the segment is significant. Note that, if the trigger signal is equal to the level for a period of time, all those points are possible triggers.

1.2.2 Selecting segments

The segments chosen to be displayed are disjoint in time.

$$
\begin{array}{l}
\mathit{SegPartition} == \{ sp : \mathbb{P}\, \mathit{Segment} \mid \\
\quad (\forall s1, s2 : sp \bullet s1 \neq s2 \Rightarrow \operatorname{dom} s1 \cap \operatorname{dom} s2 = \{\})\}
\end{array}
$$

Each segment displayed must be related to a trigger point: it should commence at a given delay offset from the trigger point and be of a given duration. In addition, as oscilloscopes need a recovery period after displaying each segment there can be a gap after each segment in which trigger points may be ignored; this means that the actual segments displayed are not uniquely determined. The maximum gap is a fixed for our oscilloscope at 1 millisecond.

$$Gap == (0.001, Second)$$

In *SelectSegs* the partition *part* is used to specify that total segments including both the delay and duration do not overlap, and the partition *segments* represents the same segments but ignoring the delay portion of each.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{SelectSegs}}\\
\hline
triggerpoints : \mathbb{P}\ Time \\
channel : DiffSignal \\
Delay, Duration : Time \\
segments : SegPartition \\
\hline
(\exists\, part : SegPartition \bullet \\
\quad (\forall\, seg : part \bullet \\
\quad\quad inf(\mathrm{dom}\, seg) \in triggerpoints \wedge \\
\quad\quad size(\mathrm{dom}\, seg) = Delay + Duration) \wedge \\
\quad \neg\, (\exists\, st : ContigPeriod \bullet \\
\quad\quad st \cap \mathrm{dom}(\bigcup part) = \{\} \wedge \\
\quad\quad (\exists\, t : st \cap triggerpoints \bullet Gap < t - inf(st))) \wedge \\
\quad segments = \{seg : part \bullet \\
\quad\quad \{t : \mathrm{dom}\, seg \mid inf(\mathrm{dom}\, seg) + Delay < t\} \lhd seg\}) \\
\hline
\end{array}
$$

1.2.3 Segment display

Each segment is displayed according to the transformations given in Section 1.1.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{DisplaySegs}} \\
\hline
segments : SegPartition \\
DisplayKnobs \\
display : Trace \\
\hline
display = \bigcup\{DisplayTrace \mid segment \in segments \bullet clipped\} \\
\hline
\end{array}
$$

The reference to *DisplayTrace* here is used to apply its constraint to each segment in the set of *segments* to be displayed and hence determine the corresponding clipped traces.

1.2.4 A single-channel, ideal oscilloscope

Putting the triggering, selection and display together we get the operation of a single-channel oscilloscope.

$$
\begin{aligned}
&IdealOscilloscope \mathrel{\widehat=} \\
&\quad (DetectTrig \land SelectSegs \land DisplaySegs) \setminus (triggerpoints, segments)
\end{aligned}
$$

2 A realistic oscilloscope

As specified above, our oscilloscope has perfect resolution both of times and signal levels. As such, it is not implementable. We need to relax our specification to allow for approximation of signal level, time skew and jitter, and bounded frequency response. As well, an oscilloscope has a limit on the maximum input voltage that it can handle safely. We begin with the frequency response limitations.

2.1 Frequency response limitations

The bounded frequency response of an oscilloscope shows up in two places in our specification. If there is a sharp spike on the input channel (at a frequency above the frequency response of the oscilloscope) it may not appear on the display. Similarly, if there is a sharp spike in the trigger input, it may not be detected as a trigger event by the oscilloscope.

Both these problems may be addressed by considering that the oscilloscope processes a signal close to the actual input, but missing the frequency components above the response frequency of the oscilloscope.

To specify this relationship we make use of a Fourier transform of the input signal, so that we can view it in the frequency domain rather than the time domain. (Our *DiffSignal* satisfies the Dirichlet conditions so a Fourier transform is well defined.) Frequencies are given in hertz.

$$Frequency == \mathbb{R} \times \{Hertz\}$$

We define the following function for describing the Fourier summation.

$$\begin{array}{l} FourierSum == (\lambda\, a, b : \mathbb{N} \longrightarrow \mathbb{R};\ period : Time \bullet \\ \quad (\lambda\, t : Time \bullet \\ \qquad \Sigma[\![n : \mathbb{N} \bullet a(n) * sin(2 * \pi * n * t \,/\, period) \\ \qquad\quad + b(n) * cos(2 * \pi * n * t \,/\, period)]\!])) \end{array}$$

To define the frequency response required of the oscilloscope we introduce a frequency response envelope: for each frequency it gives a factor (between zero and one) relating the perceived signal at that frequency to the actual signal.

$$\mid\ Response : Frequency \longrightarrow \{x : \mathbb{R} \mid 0 \leq x \leq 1\}$$

A typical response curve may indicate that the perceived signal should be within 3db of the actual at 10 gigahertz, with a 3db per decade rolloff after that frequency.

The following summarises the relationship between the perceived and actual signals in terms of the *Response* envelope. Both *realchan* and *freqchan* are single contiguous segments; we can think of them as representing the period the oscilloscope is turned on.

$$\begin{array}{|l} \multicolumn{1}{l}{FreqResponse} \\ \hline realchan, freqchan : DiffSignal \\ \hline \mathrm{dom}\ realchan = \mathrm{dom}\ freqchan \land \\ (\exists\, a, b, c, d : \mathbb{N} \longrightarrow \mathbb{R};\ period : Time \bullet \\ \quad period = size(\mathrm{dom}\ realchan) \land \\ \quad realchan = \mathrm{dom}\ realchan \lhd FourierSum(a, b, period) \land \\ \quad freqchan = \mathrm{dom}\ freqchan \lhd FourierSum(c, d, period) \land \\ \quad (\forall\, n : \mathbb{N} \bullet \\ \qquad c(n) = Response(n \,/\, period) * a(n) \land \\ \qquad d(n) = Response(n \,/\, period) * b(n))) \\ \hline \end{array}$$

Aside: The above puts a rather strict frequency response requirement on the oscilloscope. The requirement could be relaxed by use of two frequency response envelopes; the response should then lie between these two envelopes.

2.2 Over voltage limitations

The maximum voltage that may be applied on the input is limited to 50 volts.

$$MaxVoltage == (50, Volt)$$

If this is exceeded then, afterwards, we cannot assume the oscilloscope will function correctly. We model this by allowing the signal *overchan* (the output of this schema) to do as it pleases after the maximum voltage is exceeded.

OverVoltage

$freqchan, overchan : DiffSignal$

$\mathrm{dom}\, freqchan = \mathrm{dom}\, overchan$

$$\begin{aligned}
&\forall t : \mathrm{dom}\, freqchan \bullet \\
&\quad (\forall t' : \mathrm{dom}\, freqchan \bullet \\
&\quad\quad t' \leq t \Rightarrow \mathrm{abs}(freqchan(t')) \leq MaxVoltage) \\
&\quad \Rightarrow overchan(t) = freqchan(t)
\end{aligned}$$

2.3 Approximation and skew limitations

As well as the frequency response problem, we need to allow for time skew as well as inaccuracies in time and signal level displayed. These problems are more easily tackled in the time domain. The skew, jitter and signal error are parameters to the specification; the latter two are interpreted as fractional errors related to the size of the screen. For our oscilloscope the skew must be less than 100 nanoseconds and the approximation errors less than 0.1%.

$Skew : Time$

$Jitter, Error : \{x : \mathbb{R} \mid 0 \leq x \leq 1\}$

$$\begin{aligned}
&(0, Second) \leq Skew \leq (0.0000001, Second) \land \\
&0 \leq Jitter \leq 0.001 \land \\
&0 \leq Error \leq 0.001
\end{aligned}$$

The skew represents the average time delay between the input signal and its appearance on the screen, while the jitter is the maximum variation either side of this average.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{Approximation}} \\
\hline
overchan, channel : DiffSignal \\
DisplayKnobs \\
\hline
\textbf{let}\ absjitter == Jitter * HScale * HMax \bullet \\
\textbf{let}\ abserror == Error * VScale * VMax \bullet \\
\quad (\exists\, skewfn : Time \longrightarrow Time \bullet \\
\qquad skewfn \approx_{absjitter} (\lambda\, t : Time \bullet t - Skew) \land \\
\qquad channel \approx_{abserror} overchan \circ skewfn) \\
\hline
\end{array}$$

Approximation between functions is defined by the following.

$$\begin{array}{|l}
\multicolumn{1}{l}{[X]} \\
\hline
_ \approx_{e} _ : (X \mathrel{+\!\!\!\rightarrow} \mathbb{R}) \leftrightarrow (X \mathrel{+\!\!\!\rightarrow} \mathbb{R}) \\
\hline
f \approx_{e} g \Leftrightarrow \\
\quad \operatorname{dom} f = \operatorname{dom} g \land \\
\quad (\forall\, x : \operatorname{dom} f \bullet \operatorname{abs}(f(x) - g(x)) \leq e) \\
\hline
\end{array}$$

2.4 Display limitations

Depending on the type of display envisaged, additional display limitations need to be incorporated into the specification. For example, if a digital bitmap display is used, display approximations as specified in [Mar85] may be appropriate. As these limitations are adequately covered there, we do not consider them in this example specification. We also do not cover such issues as the persistence of the spot on the screen.

2.5 Combined limitations

The limitations on the input channel are the combination of the frequency response, over voltage and approximation limitations.

$$\begin{aligned}
&\mathit{SigLimits} \mathrel{\widehat{=}} \\
&\qquad (\mathit{FreqResponse} \wedge \mathit{OverVoltage} \wedge \mathit{Approximation}) \setminus \\
&\qquad\qquad (\mathit{freqchan}, \mathit{overchan})
\end{aligned}$$

The same limitations apply to the trigger input.

$$\mathit{TrigLimits} \mathrel{\widehat{=}} \mathit{SigLimits}[\mathit{realtrig}/\mathit{realchan}, \mathit{trigger}/\mathit{channel}]$$

TrigLimits is formed from *SigLimits* by renaming the variables *realchan* and *channel* to *realtrig* and *trigger*, respectively.

2.6 A single-channel oscilloscope

We are now finally in a position to specify fully a single-channel, single-trigger oscilloscope.

$$\begin{aligned}
&\mathit{Oscilloscope} \mathrel{\widehat{=}} \\
&\qquad (\mathit{IdealOscilloscope} \wedge \mathit{SigLimits} \wedge \mathit{TrigLimits}) \setminus (\mathit{channel}, \mathit{trigger})
\end{aligned}$$

Note that the ideal channel and trigger have been hidden in this level of specification. They were introduced so that we could separate the concerns of what we ideally want an oscilloscope to do, and the unavoidable limitations of actually implementing an oscilloscope; they cannot be considered to be part of the observable behaviour of an oscilloscope.

3 Improvements and extensions

The knobs that control the display (*Hscale*, *VScale*, *HOffset*, *VOffset*, *Level*, *Slope*, *Delay* and *Duration*) were chosen to be fixed to simplify the above presentation. This is unrealistic but can be overcome by making each of these variables a function of time and changing every reference to one of these variables so that it is indexed by time. Another approach would be to consider the knobs as being fixed at the values current at the start of each trace; this approach, however, would not model the behaviour of many oscilloscopes where, for example, if the vertical offset is changed during the display of the trace, this is clearly visible at horizontal scales of, say, one second per centimetre. Another minor problem with the control knobs is that we have not specified bounds on their values; this is unrealistic, but easily remedied.

There are many other features that we could have added to our oscilloscope, but we have kept to a simple example in order to convey the basic principles. For example, the specification as presented assumes DC coupling of the input signal; AC coupling could be included as a high-pass filter in a way similar to the treatment of frequency response as a low-pass filter. Such an additional facility could be added to the oscilloscope specification in a manner similar to the way scaling and translation are handled: an additional schema describing the facility could be defined and used within *DisplayTrace*.

Multiple channels could be added to the oscilloscope specification by using the current building blocks to describe the operation of a single channel and then including an array of such channels at the next stage of the specification; the combined display can be defined as the union of the displayed traces of the individual channels. The model chosen for a trace — a function from time to a position on the screen — also allows for the display of one channel versus another: at any time t the X voltage (Y voltage) is converted to the X coordinate (Y coordinate) of the trace at time t.

4 Conclusions

By specifying an ideal oscilloscope first, we can concentrate on what the user would ideally like from the oscilloscope, before going on to consider physical limitations. This allows one to use a simpler ideal model to specify the basic facilities required. The specification of the ideal oscilloscope makes use of the real number continuum to model the signal and trigger inputs, as well as the displayed trace. This leads to some simplifications when compared to using discrete models, as methods from real analysis, such as derivatives, can be used. The ideal oscilloscope is specified via two major components covering, respectively, how a single trace is displayed on the screen, and how segments of the input signal are selected for display according to the trigger input.

To allow implementation the ideal specification is relaxed to allow for physical limitations. This is done by imagining that the ideal oscilloscope processes signal and trigger inputs that are approximations to the actual inputs. These approximations allow for bounded frequency response, over-voltage limitations, inaccuracies in the signal level, and time skew and jitter. It is clear that this is not a suitable approach to structuring an implementation. The inputs to the ideal oscilloscope are used purely as a descriptive technique and do not correspond to any directly observable part of an implementation. However, this technique allows us to separate the concerns of what is ideally required by the user and what approximations are allowed for implementation; hence this technique is well suited our purpose here of providing a clear specification.

The basic components of the specification were written as Z schemas. These components were then combined, using schema conjunction, to build up the structure of the specification. The use of schemas as building blocks presents a stylistic departure from the use of functions as building blocks in [DG90, DG88]. This departure is, in part, motivated by a desire to have a consistent approach to specifying components whether they be deterministic — the output being uniquely determined by the input — or non-deterministic — there being a number of possible outputs for a given input. While the basic transformations such as scaling and translation are deterministic, the selection of segments to be displayed (*SelectSegs*) and the approximations used to make the specification realistic (Section 2) are non-deterministic and hence cannot be simply specified via functions.

The use of physical units throughout the specification was found to be an aid (similar to type checking in programming languages) in creating a consistent specification. Calculating the physical dimensions of expressions and checking for consistency provided a simple, but useful, consistency check and is to be recommended in specifications involving physical quantities.

Acknowledgements

This work was completed while I was visiting the Programming Research Group of Oxford University and I would like to acknowledge the hospitality of the Programming Research Group and Wolfson College, and the financial assistance of the Special Studies Program of the University of Queensland.

I would like to acknowledge the work of David Garlan and Norman Delisle

from Tektronix; this paper attempts to extend their excellent work. My thanks also go to Brendan Mahony, Carroll Morgan, Mike Spivey and members of IFIP Working Group 2.3 for useful comments on earlier drafts of this paper.

References

[DG88] Norman Delisle and David Garlan. Formally specifying electronic instruments. Technical Report CR-88-11, Computer Research Laboratory, Tektronix, Inc., P.O. Box 500, Beaverton, Oregon OR 97077, October 1988. Draft.

[DG90] Norman Delisle and David Garlan. A formal specification of an oscilloscope. *IEEE Software*, 7(5):29–36, September 1990.

[Hay93] I. J. Hayes, editor. *Specification Case Studies.* Prentice Hall International, second edition, 1993.

[Mar85] Lynn S. Marshall. A formal specification of line representations on graphics devices. In *Lecture Notes in Computer Science – 186*, pages 129–147. Springer Verlag, March 1985.

[Spi92] J. M. Spivey. *The Z Notation: A Reference Manual.* Prentice Hall International, second edition, 1992.

A Notation

Logic We make use of the logical operators $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\Rightarrow$ (implies) and $\Leftrightarrow$ (if and only if), as well as the logical quantifiers $\forall$ (for all) and $\exists$ (there exists). The quantified variables range over an explicitly specified type:

$$\begin{aligned}(\exists\, x : T \bullet P) &\Leftrightarrow (\exists\, x \bullet x \in T \wedge P),\\ (\forall\, x : T \bullet P) &\Leftrightarrow (\forall\, x \bullet x \in T \Rightarrow P).\end{aligned}$$

The syntax "$x : T$" is used consistently throughout the notation to denote the introduction of a new variable x of type T; we distinguish it from the notation "$x \in T$" which is always a predicate.

Definitions The symbol "==" is used to denote "is defined to be". Local definitions within predicates are introduced by the keyword **let** :

$$(\mathbf{let}\ x == t \bullet P) \Leftrightarrow (\exists\, x \bullet x = t \wedge P).$$

Sets A set can be defined by enumerating its elements, e.g., $\{1, 4, 9\}$, or via a predicate that is satisfied by exactly the members of the desired set. For example, the even natural numbers can be specified by

$$\{n : \mathbb{N} \mid (\exists\, k : \mathbb{N} \bullet 2 * k = n)\},$$

where $\mathbb{N}$ is the set of all natural numbers. A variant of the above notation allows a term to be specified as well. For example, the set of squares of even natural numbers can be specified by

$$\{n : \mathbb{N} \mid (\exists\, k : \mathbb{N} \bullet 2 * k = n) \bullet n * n\}.$$

The slightly unconventional syntax which puts the term $n * n$ at the end rather than the beginning is chosen for consistency with other Z notation (see, for example, λ notation below).

We use the following conventional set operators: $\in$ (membership), $\cup$ (union), $\cap$ (intersection), $\times$ (Cartesian product), $\mathbb{P}$ (powerset — set of all subsets of its operand), $\mathbb{P}_1$ (non-empty subsets), and $\bigcup$ (generalised union — when applied to a set of sets it forms the union of all the sets).

Real numbers The set of all real numbers is denoted $\mathbb{R}$ and the non-negative reals by $\mathbb{R}_+$. We assume all the usual operators on reals, including *abs* for absolute value.

Functions In Z, functions are represented as sets of pairs such that each first component is associated with a unique second component. The domain of definition of a function f, denoted $\operatorname{dom} f$, is the set of all first components of the pairs forming the function, and the range, denoted $\operatorname{ran} f$, is the set of all second components. A partial function f from X to Y is declared by $f : X \rightarrow\!\!\!\!\!+ Y$, while a total function g is declared by $g : X \longrightarrow Y$. A total function is defined for all values in X, that is, $\operatorname{dom} g = X$.

We use a typed version of lambda notation. Given an argument x of type T, the function $(\lambda\, x : T \bullet t)$ returns the value of the term t. The form

$(\lambda\, x : T \mid P \bullet t)$ introduces a constraint on the arguments that the function accepts: they must satisfy the predicate P. We have

$$\mathrm{dom}(\lambda\, x : T \mid P \bullet t) = \{x : T \mid P\}.$$

Given a function $f : X \rightarrowtail\!\!\!\!\!\!\;+ Y$ and sets $S : \mathbb{P}\, X$ and $T : \mathbb{P}\, Y$, the term $S \lhd f$ is the function f domain restricted to the set S (it contains just those pairs in f whose first components are in S); and the term $f \rhd T$ is the function f range restricted to T (it contains just those pairs in f whose second components are in T).

Summation The notation $\Sigma[\![x : T \bullet t]\!]$ denotes the sum of terms t, for x ranging over all the values in T.

Schemas A schema consists of a set of declarations of variables plus a predicate constraining those variables; it is written as a box with the declarations above the middle line and the predicate below. The name of the schema is on the top of the box.

Schema conjunction ($\wedge$) can be used to combine two schemas to form a new schema; the declarations of the two schemas are merged (variables with the same name must have identical types) and predicates are conjoined. For example, *Translate* $\wedge$ *Clip* is the following schema.

Translate_Clip

$$\begin{array}{l}
\mathit{scaled} : \mathit{Trace} \\
\mathit{HOffset}, \mathit{VOffset} : \mathit{Coord} \\
\mathit{moved} : \mathit{Trace} \\
\mathit{clipped} : \mathit{Trace} \\
\hline
\mathit{moved} = (\lambda\, t : \mathrm{dom}\, \mathit{scaled} \bullet \\
\qquad (\mathit{first}(\mathit{scaled}(t)) + \mathit{HOffset}, \mathit{second}(\mathit{scaled}(t)) + \mathit{VOffset})) \wedge \\
(\mathbf{let}\ \mathit{screen} == \\
\qquad \{x, y : \mathit{Coord} \mid 0 < x < \mathit{HMax} \wedge -\mathit{VMax} < y < \mathit{VMax}\} \bullet \\
\qquad\qquad \mathit{clipped} = \mathit{moved} \rhd \mathit{screen})
\end{array}$$

Schema variable hiding ($\setminus$) removes the named variables from the declaration part of the schema and existentially quantifies them in the predicate.

For example, $DisplayTrace$ is defined as

$$(Scale \land Translate \land Clip) \setminus (scaled, moved),$$

and expands to the following schema.

$$
\begin{array}{|l}
\hline
\textit{Scale\_Translate\_Clip} \\
segment : Segment \\
HScale : \mathbb{R}_+ \times \{Second \mathbin{/} Metre\} \\
VScale : \mathbb{R} \times \{Volt \mathbin{/} Metre\} \\
HOffset, VOffset : Coord \\
clipped : Trace \\
\hline
\exists\, scaled, moved : Trace \bullet \\
scaled = (\lambda\, t : \operatorname{dom} segment \bullet \\
\quad ((t - inf(\operatorname{dom} segment)) \mathbin{/} HScale, segment(t) \mathbin{/} VScale)) \land \\
moved = (\lambda\, t : \operatorname{dom} scaled \bullet \\
\quad (first(scaled(t)) + HOffset, second(scaled(t)) + VOffset)) \land \\
(\mathbf{let}\ screen == \\
\quad \{x, y : Coord \mid 0 < x < HMax \land - VMax < y < VMax\} \bullet \\
\qquad clipped = moved \rhd screen) \\
\hline
\end{array}
$$

The predicate part of this schema can be transformed to an equivalent predicate without the existential quantifier, at some cost in readability.

Variables declared in a schema may be renamed. For example,

$$SigLimits[realtrig/realchan, trigger/channel]$$

stands for the schema $SigLimits$ with the declarations of $realchan$ and $channel$ plus every (free) occurrence of $realchan$ and $channel$ in the predicate replaced by the variables $realtrig$ and $trigger$, respectively.

Upper and lower bounds Below we define lower/upper bounds (lb/ub), greatest/least lower/upper bounds (inf/sup), and size of a contiguous time period. Both lb and ub are relations ($\leftrightarrow$) between contiguous time periods and their corresponding bounds; when such named relations are used as infix operators they are underlined.

$$
\begin{array}{|l}
lb, ub : ContigPeriod \leftrightarrow Time \\
inf, sup : ContigPeriod \longrightarrow Time \\
size : ContigPeriod \longrightarrow Time \\
\hline
\forall st : ContigPeriod \bullet \\
\quad (\forall t : Time \bullet \\
\quad\quad (st \mathbin{\underline{lb}} t \Leftrightarrow (\forall t1 : st \bullet t \leq t1)) \land \\
\quad\quad (st \mathbin{\underline{ub}} t \Leftrightarrow (\forall t1 : st \bullet t \geq t1))) \land \\
\quad (st \mathbin{\underline{lb}} inf(st) \land \neg (\exists t1 : Time \bullet st \mathbin{\underline{lb}} t1 \land inf(st) < t1)) \land \\
\quad (st \mathbin{\underline{ub}} sup(st) \land \neg (\exists t1 : Time \bullet st \mathbin{\underline{ub}} t1 \land t1 < sup(st))) \land \\
\quad size(st) = sup(st) - inf(st)
\end{array}
$$

B Physical Units

This appendix outlines an approach to formalising physical units. The basic units can be represented by enumeration (similar to a Pascal enumerated type). The following are sufficient for this specification.

$$Unit ::= metre \mid second \mid volt$$

To represent physical quantities we need not only basic units but dimensionality. We can represent dimensionality by associating an integer with a basic unit.

$$Units == Unit \longrightarrow \mathbb{Z}$$

For example, velocity can be represented as follows.

$$velocity == \{metre \mapsto 1, second \mapsto -1, volt \mapsto 0\}$$

A quantity with zero dimensionality in all units has type $Null$.

$$Null == \{u : Unit \bullet u \mapsto 0\}$$

For our specification the following abbreviations are useful. The function override operator ($\uplus$) in the following definitions has the effect of replacing the dimensionalities of its first operand with those of its second operand.

$$
\begin{array}{l}
Metre == Null \uplus \{metre \mapsto 1\} \\
Second == Null \uplus \{second \mapsto 1\} \\
Volt == Null \uplus \{volt \mapsto 1\}
\end{array}
$$

Useful functions for constructing units from other units are the following.

$$
\begin{array}{|l}
\_ * \_ : \mathit{Units} \times \mathit{Units} \longrightarrow \mathit{Units} \\
\_^{\sim} : \mathit{Units} \longrightarrow \mathit{Units} \\
\_ / \_ : \mathit{Units} \times \mathit{Units} \longrightarrow \mathit{Units} \\
\hline
us1 * us2 = (\lambda\, u : \mathit{Unit} \bullet us1(u) + us2(u)) \\
us^{\sim} = (\lambda\, u : \mathit{Unit} \bullet -(us\ u)) \\
us1 \,/\, us2 = us1 * (us2^{\sim})
\end{array}
$$

The following abbreviation is useful for our specification.

$$\mathit{Hertz} == \mathit{Second}^{\sim}$$

Real numbers are extended with units of dimension. The magnitude and units of such a quantity can be extracted with *magn* and *units*.

$$
\begin{array}{l}
\mathbb{R} == \mathbb{R} \times \mathit{Units} \\
\mathit{magn} == \mathit{first}[\mathbb{R}, \mathit{Units}] \\
\mathit{units} == \mathit{second}[\mathbb{R}, \mathit{Units}]
\end{array}
$$

Normal arithmetic operations on reals can be upgraded to work on physical quantities; to be well defined, these may require the units to be consistent.

$$
\begin{array}{|l}
- : \mathbb{R} \longrightarrow \mathbb{R} \\
\mathrm{abs} : \mathbb{R} \longrightarrow \mathbb{R} \\
\_ + \_ : \mathbb{R} \times \mathbb{R} \nrightarrow \mathbb{R} \\
\_ - \_ : \mathbb{R} \times \mathbb{R} \nrightarrow \mathbb{R} \\
\_ * \_ : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R} \\
\_ / \_ : \mathbb{R} \times \mathbb{R} \nrightarrow \mathbb{R} \\
\_ < \_ : \mathbb{R} \leftrightarrow \mathbb{R} \\
\_ \leq \_ : \mathbb{R} \leftrightarrow \mathbb{R} \\
\_ \geq \_ : \mathbb{R} \leftrightarrow \mathbb{R} \\
\hline
-(v, u) = (-v, u) \\
\mathrm{abs}(v, u) = (abs(v), u) \\
\mathrm{dom}(\_ + \_) = \{vu1, vu2 : \mathbb{R} \mid units(vu1) = units(vu2)\} \\
(v1, u) + (v2, u) = (v1 + v2, u) \\
\mathrm{dom}(\_ - \_) = \{vu1, vu2 : \mathbb{R} \mid units(vu1) = units(vu2)\} \\
(v1, u) - (v2, u) = (v1 - v2, u) \\
(v1, u1) * (v2, u2) = (v1 - v2, u1 * u2) \\
\mathrm{dom}(\_ / \_) = \{vu1, vu2 : \mathbb{R} \mid magn(vu2) \neq 0\} \\
v2 \neq 0 \Rightarrow (v1, u1) / (v2, u2) = (v1 / v2, u1 / u2) \\
(v1, u) < (v2, u) \Leftrightarrow v1 < v2 \\
(v1, u) \leq (v2, u) \Leftrightarrow v1 \leq v2 \\
(v1, u) \geq (v2, u) \Leftrightarrow v1 \geq v2
\end{array}
$$