A Continuous ASM Modelling Approach to Pacemaker Sensing

RICHARD BANACH, University of Manchester
HUIBIAO ZHU, East China Normal University
WEN SU, Shanghai University
XIAOFENG WU, East China Normal University

The cardiac pacemaker system, proposed as a problem topic in the Verification Grand Challenge, offers a range of difficulties to address for formal specification, development and verification technologies. We focus on the sensing problem, the question of whether the heart has produced a spontaneous heartbeat or not. This question is plagued by uncertainties arising from the often unpredictable environment that a real pacemaker finds itself in. We develop a time domain tracking approach to this problem, as a complement to the usual frequency domain approach most frequently used. We develop our case study in the continuous ASM (Abstract State Machine) formalism, which is briefly summarised, through a series of refinement and retrenchment steps, each adding new levels of complexity to the model.

Categories and Subject Descriptors: F.3.1 [**Specifying and Verifying and Reasoning about Programs**]: Specification techniques

General Terms: Verification, Theory, Design, Algorithms

Additional Key Words and Phrases: Continuous ASM, Rigorous Design and Development, Cardiac Pacemakers, Sensing.

ACM Reference Format:
Richard Banach, Huibiao Zhu, Wen Su and Xiaofeng Wu, 2013. A Continuous ASM Modelling Approach to Pacemaker Sensing. *ACM Trans. Embedd. Comput. Syst.* 9999, 8888, Article 77777777 (Month 2013), 36 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The Verification Grand Challenge [Jones et al. 2006; Woodcock 2006; Woodcock and Banach 2007] has proposed a number of case study problems to test the capabilities of a variety of formal development and verification approaches. Among these is the cardiac pacemaker problem, crystallised in the Boston Scientific documentation [Boston Scientific 2007]. While the focus of [Boston Scientific 2007] is on the various 'modes of operation' of a typical pacemeker (taking as settled the question of whether the pacemaker needs to stimulate the heart during any period, corresponding to whether it has sensed a heartbeat or not), we take a variant view of the challenge problem, and focus on the sensing problem itself. This question is plagued by uncertainties arising from the often unpredictable environment that a real pacemaker finds itself in. Our aim is to see how our formal framework deals with such a poorly constrained problem.

Pacemakers are interesting from this viewpoint since the historical approach to pacemaker evolution and development is so deeply ingrained in very pragmatically based clinical practice and exper-

A portion of the work reported in this paper was done while R. Banach was a visiting researcher at the Shanghai Key Laboratory of Trustworthy Computing at East China Normal University. The support of ECNU is gratefully acknowledged. Huibiao Zhu is supported by National High Technology Research and Development Program of China (No. 2012AA011205), National Natural Science Foundation of China (No. 61361136002 and No. 61321064), Shanghai Knowledge Service Platform Project (No. ZF1213) and Shanghai Minhang Talent Project. Author's addresses: Correspondence to R. Banach, School of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K.; W. Su, School of Computer Engineering and Science, Shanghai University, Shanghai, P.R. China; H. Zhu and X. Wu, Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, 3663 Zhongshan Road North, Shanghai 200062, P.R. China.

DOI:http://dx.doi.org/10.1145/0000000.0000000

imentation [Barold et al. 2010; Ellenbogen and Wood 2008; Ellenbogen et al. 2006; Kusumoto and Goldschlager 2007]. From the formal development standpoint, the most obvious and pressing question that this prompts is: ‘what exactly are the requirements?’ Any formal approach to pacemaker behaviour brings with it a sharpening of some aspects of requirements, whether explicit or implicit.

Most formal approaches that deal with systems which are hybrid (like the pacemaker), tackle a model of the system at a single level of abstraction. A comprehensive survey of existing techniques can be found in [Carloni et al. 2006]. In contrast, our approach is via a formalism that allows top-down development, the aim being to see if the layered approach helps in systematising in a convincing way all the aspects of uncertainty that the modelling encounters. For this purpose we use Continuous ASM, a continuous extension of the well known discrete Abstract State Machine (ASM) formalism [Börger and Stärk 2003; Börger 2003]. Continuous ASM combines discrete and continuous transitions, as do all frameworks for hybrid systems, but in addition, inherits a mature refinement concept, which it extends to the continuous domain.

The rest of this paper is structured as follows. In Section 2 we set the scene by briefly introducing the environment of pacemakers, work done to date, and our own focus: sensing. In Section 3 we review the essentials of Continuous ASM, pointing elsewhere for a fully detailed treatment. Section 4 covers the relevant top-down machinery. Section 5 then explores the sensing problem in more detail. Section 6 which follows, is an extensive case study on the sensing problem, presenting a series of ASM models for sensing which start with a simple reference model, and become increasingly algorithmic. The formal relationships between these models are discussed. Section 7 looks back and draws general lessons and principles that should prove useful in other formal developments of scientific applications. Section 8 discusses some related technical issues. Section 9 concludes.

2. HEART PACEMAKERS, THE PACEMAKER CHALLENGE, AND SENSING

Everyone has a heart, and every reader of this paper will have a reasonable idea of how the heart works. The heart has two atria and two ventricles. Blood collects in the atria, and is decanted into the ventricles by a wave of muscular contraction in response to an electrical impulse from the sinoatrial node. A short time later, another electrical impulse, relayed from the atrial focus by a nervous system pathway, causes a powerful wave of contraction in the much more heavily muscled ventricles to pump the blood round the body. In this manner the circulation of the blood round the body is maintained.

The electrical activity is basically the same in both of these cases of muscular contraction. The cell boundaries of the muscle fibres making up the myocardium are normally polarized, with positive charges on the outside, and negative charges on the inside. The electrical impulse delivered by the nervous system causes a co-ordinated wave of depolarization to spread through the myocardium, which produces the muscular contraction and hence the pumping effect.

The heart beats when it is told to do so by its external environment. In this case it is the sinoatrial node, with its pacing activity modulated by the vagus nerve system, that initiates atrial depolarization. In normal working, the atrial pulse has to be of the right strength and characteristics to cause depolarization. Similarly, the ventricular pulse, trasmitted to the atrioventricular node via a specialised pathway, has to arrive at the right time relative to the atrial pulse to cause the ventricular depolarization to happen, and to cause blood to be pumped round the body in the most hemodynamically efficient way. Inevitably, various things can go wrong with the nervous mechanisms that cause all this to happen. Problems of different kinds can arise with atrial depolarization, with ventricular depolarization, with both, and with the relationship between them. Such deficiencies are collectively referred to as heart block, with different types of block indicating various specific deficiencies. Heart block can be addressed by the implantation and suitable configuring of pacemakers, whose aim is to supply electrical pulses that substitute for ones that the body is unable to generate properly itself.

To work effectively, pacemakers require a remarkable and productive synergy between medical practice and research, electronics, devices, materials science, and more recently computing; this was remarked on as early as 1974 [Goldman et al. 1974]. Over the years, pacemakers have evolved into very sophisticated devices. To facilitate research into the computing dimension of pacemaker

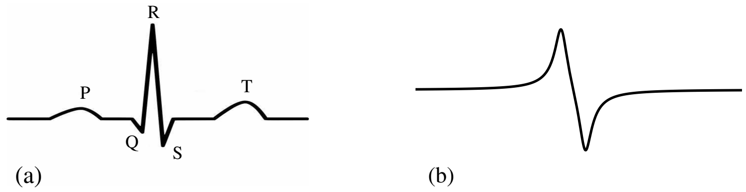

Fig. 1. (a) Schematic of a typical normal surface electrocardiogram (ECG) of a full heartbeat, showing the ‘P wave’ (atrial depolarization), the ‘QRS complex’ (ventricular depolarization), and the ‘T wave’ (repolarization). (b) Schematic of a typical normal internal cardiac electrogram corresponding to a single depolarization (either atrial or ventricular).

technology, a public domain specification of a pacemaker system has been produced by Boston Scientific [Boston Scientific 2007] for use in the Verification Grand Challenge [Jones et al. 2006; Woodcock 2006; Woodcock and Banach 2007]. This has been prompted by the ever increasing complexity of contemporary designs, in turn spurred by the ever decreasing size and power requirements of contemporary VLSI technology.

A perusal of [Boston Scientific 2007] reveals that for it to be useful, the reader has to rely on a very large amount of additional knowledge. In this regard it resembles the majority of requirements and specification documents in the industrial sphere, which are written without regard to the level of completeness that would make them truly viable for use as a starting point in reliable independent system development.

The most self-contained part of [Boston Scientific 2007] deals with the different ‘modes’ of pacemaker working. The wide range of possible electrical stimulation defects indicated above gives rise to a corresponding range of modes of pacemaker operation, each designed to address a specific defect. Each of these modes is assembled out of a number of available pacemaker features, where each relevant feature of a mode is active for a given duration specified in milliseconds, and is tunable by the physician within a given range. Aided by a moderate amount of supplementary information available in the cited literature, this aspect of pacemaker operation becomes relatively tractable, and has attracted a certain amount of interest from the verification community [Macedo et al. 2008; Gomes and Oliveira 2009; 2010; Méry and Singh 2009; 2010; 2011].

Unless a pacemaker is working in an ‘asynchronous’ mode, in which it takes no notice of the heart’s spontaneous behaviour, the pacemaker listens to the heart’s innate activity. When it detects that the heart has managed to generate a suitable depolarization unaided, it suppresses the generation of an artificial pulse — this functionality is known as sensing. The heart’s electrical activity can be detected on the patient’s skin (via an electrocardiogram, (ECG)), and inside the heart itself (via a cardiac electrogram (EG)). Fig. 1 illustrates the distinction between these, showing schematics of the typical form of an ECG, and of an individual pulse of an EG corresponding to a single depolarization. Since the pacemaker is implanted inside the patient, it is the cardiac electrogram that it must sense — and the telltale signal that it seeks to identify is the signature of depolarization, which looks like Fig. 1.(b) (though various distortions of this basic shape are possible, which we will ignore in this paper).

In the clinical view of sensing, e.g. in [Schuchert et al. 2005; Aubert et al. 1989; Keinert et al. 1979; DeCaprio et al. 1977; Furman et al. 1977], it is always assumed that sensing is done by a filter circuit. This works in the frequency domain, and registers electrical activity in the heart in the relevant frequency range, while suppressing inappropriate frequencies. The circuit outputs a discrete ‘Yes’ or ‘No’ signal if the power spectrum of the filtered input is strong enough in the relevant frequency range.

The majority of the clinical and computer science literatures in this area simply assume that it is known by the pacemaker whether or not an innate pulse has occured, and go on from there. Nevertheless, the clinical literature always cautions that there is a significant risk of sensing circuits confusing a true depolarization with other electrical activity, whether arising in the heart, in nearby tissue, or outside the body. The source of this problem is the following.

The fact that the filter circuit senses only the power spectrum, means that the phase information in the signal is discarded. Consequently, many signals having essentially the right power spectrum, but with a shape very different from Fig. 1.(b), can be erroneously identified with a depolarization if their phase distribution differs too drastically from that of Fig. 1.(b). More detail on sensing can be found in Chapter 3 of [Ellenbogen et al. 2006].

In this paper, we develop a time domain tracking of the putative impulse. This seeks to distinguish pulses of electrical activity arising from genuine depolarizations from spurious other signals sharing a similar power spectrum, by looking at their real time shape. Since pacemakers are capable of measuring the internal cardiac electrogram, and the public specification [Boston Scientific 2007] explicitly stipulates such a capability, this is an entirely credible strategy to investigate. Such a system could usefully supplement the frequency domain sensing strategy discussed above.

3. ASM, DISCRETE AND CONTINUOUS

In this section we briefly review the essentials of Continuous ASM, which builds on the foundations of traditional ASM [Börger and Stärk 2003; Börger 2003]. A more detailed account of Continuous ASM can be found in [Banach et al. 2013].

3.1. Continuous ASM Models

To keep things simple, we assume that the states of an ASM model are given by valuations of the tuple of variables relevant to the model, i.e. functions from the tuple of variables to the tuple of the variables' types. Although this restricts the generality of the ASM formalism, it is sufficient for our case study. To extend such models to include continuously varying phenomena, we partition the variables into two: the **mode variables**, whose types are discrete sets, and which can only change discontinuously and discretely, and the **pliant variables**, whose types include topologically dense sets, and which are permitted to evolve both continuously and via discrete changes.

We model time as an interval $\mathcal{T}$ of the real numbers $\mathbb{R}$, with a finite left endpoint for the time at which the initial state of the model is created, and with a right endpoint which is either finite or infinite, depending on the duration of the dynamics. The values of all variables thus become functions of $\mathcal{T}$. For the mode variables, this function is deemed to be a piecewise constant function, constant on each element of a sequence of left-closed right-open intervals, whence $\mathcal{T}$ itself also partitions into a sequence of left-closed right-open intervals, $\langle [t_0 \dots t_1), [t_1 \dots t_2), \dots \rangle$, the coarsest partition of $\mathcal{T}$ such that all discontinuous changes take place at some boundary point t_i.

In a typical interval $[t_i \dots t_{i+1})$, the mode variables will be constant, but the pliant variables will change continuously. However, merely insisting on continuity still allows for a wide range of mathematically pathological behaviours. To eliminate these, we make the following restrictions:

I Zeno: there is a constant δ_{Zeno}, such that for all i needed, $t_{i+1} - t_i \geq \delta_{\mathsf{Zeno}}$.[1]

II Limits: for every variable x, and for every time $t \in \mathcal{T}$, the left limit $\lim_{\delta \to 0} x(t - \delta)$, written $\overrightarrow{x(t)}$, and the right limit $\lim_{\delta \to 0} x(t + \delta)$, written $\overleftarrow{x(t)}$, (with $\delta > 0$ in each case) both exist, and for every t, $x(t) = \overleftarrow{x(t)}$. (N. B. At the endpoint(s) of $\mathcal{T}$, any missing limit is defined to equal its counterpart.)

III Differentiability: The behaviour of every pliant variable x in the interval $[t_i \dots t_{i+1})$ is given by the solution of a well posed initial value problem $\mathcal{D}\,xs = \phi(xs, t)$ (where xs is a relevant tuple of pliant variables and $\mathcal{D}$ is the time derivative). 'Well posed' means that $\phi(xs, t)$ has Lipschitz constants which are uniformly bounded over $[t_i \dots t_{i+1})$ bounding its variation with respect to xs, and that $\phi(xs, t)$ is measurable in t.

[1] Much has been said about the Zeno problem, and many formulations of Zeno restrictions are possible. For us, the most important observation is that proving Zeno-freedom in a system usually requires global knowledge of the system's reachabilty relation. As such, it is not easy to demand Zeno-freedom statically and so **I** is a goal rather than a static restriction.

It is recognised that ASM types can be mathematically complex entities. Therefore it is intended that **I-III** above apply to variables with as general a type as might be needed, provided that the concepts required in **I-III** (such as left/right limits, initial value problem, Lipschitz constants, uniform boundedness, measurability) make sense for them.

With **I-III** in place, the behaviour of every pliant variable is piecewise absolutely continuous, with the variation being described by a suitable differential equation (DE).

Accompanying the distinction between mode and pliant variables, is a distinction between mode and pliant transitions. Mode transitions are just like conventional ASM transitions in that they record a discrete transition from before-values to after-values of the mode variables, albeit that these are the values of piecewise constant functions of time. A rule for a mode transition OP can be written using familiar ASM notation as follows, where in addition, we have decorated the variables with overarrows to indicate which limit is to be used in evaluating them at runtime when the rule is executed. Stripping off the overarrows from (1) yields the syntactic form one would write for a specific application.

$$
\begin{aligned}
&\textsc{Op}(\mathbf{in}\ \overrightarrow{is}, \mathbf{out}\ \overleftarrow{os}) \ = \\
&\mathbf{if}\ \ guard(\overrightarrow{xs}, \overrightarrow{is})\ \ \mathbf{then\ choose}\ \ \overleftarrow{xs'}, \overleftarrow{os}\ \ \mathbf{with}\ \ rel(\overleftarrow{xs'}, \overrightarrow{xs}, \overrightarrow{is}, \overleftarrow{os}) \\
&\quad \mathbf{do}\ \ xs, os\ := \ \overleftarrow{xs'}, \overleftarrow{os}
\end{aligned} \tag{1}
$$

In (1) we single out *is* and *os*, the inputs and outputs (read-only and write-only respectively), while *xs* are the state variables (accessed in read/write manner). Note the double decoration of the after-state variables $\overleftarrow{xs'}$. The prime corresponds to the usual syntactic decoration that one would expect to use in distinguishing before-states (unprimed) from after-states (primed), whereas the overarrow indicates the relevant limit.

Note that the choice of left limit for before-values and right limit for after-values makes (1) into the kind of instantaneous transition that we would expect. As usual, if the after-values for *xs* and *os* are available explicitly, the relevant expression can be assigned directly in the **do** clause, and the **choose** and **with** clauses can be omitted.

Pliant transitions do the corresponding job for pliant variables. While a mode transition is a single before-/after-value pair, a pliant transition is a family of before-/after-value pairs parameterized by the relevant time interval $[t_i \dots t_{i+1})$. Moreover, instead of the change from before-values to after-values taking place instantaneously, the before-value refers to the initial value at t_i (which, by **II**, equals the right limit at t_i), while the after-value refers to an arbitrary time in the interval, so the before-value and after-value are separated in time. To reflect the constraints that apply to pliant transitions, we write rules for them thus:

$$
\begin{aligned}
&\textsc{PliOp}(\mathbf{in}\ is(t \in (t_{\mathrm{L}(t)} \dots t_{\mathrm{R}(t)})), \mathbf{out}\ os(t \in (t_{\mathrm{L}(t)} \dots t_{\mathrm{R}(t)}))) \ \stackrel{\mathrm{c}}{=} \\
&\mathbf{if}\ \ IV(xs(t_{\mathrm{L}(t)})) \wedge guard(xs(t_{\mathrm{L}(t)}))\ \ \mathbf{then\ with}\ \ rel(xs, is, os, t) \\
&\quad \mathbf{do}\ \ xs(t), os(t)\ := \ \mathbf{solve}\ DE(xs(t), is(t), os(t), t)
\end{aligned} \tag{2}
$$

In (2), the symbol $\stackrel{\mathrm{c}}{=}$ signals the presence of a pliant transition, distinguishing it from the instantaneous kind. The inputs *is* and outputs *os* are continuously absorbed from and emitted to the environment, as indicated in the signature. For an arbitrary t, we let $\mathrm{L}(t) = \max\{i \,|\, t_i \leq t\}$ and $\mathrm{R}(t) = \min\{i \,|\, t_i > t\}$ so that: (a) we do not have to know the index i explicitly in $[t_i \dots t_{i+1})$; (b) we can refer to the beginning and end of the interval during which the pliant event runs in a generic manner in the syntactic description of the event. Note that the initial values *IV* and guard *guard* depend only on the before-value of the state,[2] and not on the input, whereas *rel*, which expresses any additional constraints that must hold beyond the differential equation *DE* itself, can depend on all state and input values from the start of the interval $t_{\mathrm{L}(t)}$ up to the current time t. The assignment

[2]Normally, we would expect *IV* to depend on the pliant variables and *guard* to depend on the mode variables, but there is no need to insist on this formally.

in (2) says that the after-state and output at t should satisfy the differential equation DE (as well as rel). As for the instantaneous case, if the continuous functions of t to be assigned to xs, os are known explicitly, we can omit the **with** and/or **solve** clauses as approporiate, and just assign directly xs, os to the relevant expression.

As was the case with (1), in (2) we have adapted the syntax compared with what one would expect to see in a specific application. Thus, since the symbol $\overset{c}{=}$ signals a pliant transition already, many things, such as the time dependence of all variables, are already implicit. So the '$t \in (t_{L(t)} \dots t_{R(t)})$' in the declaration of the input '**in** *is*' is, strictly speaking, redundant, and the same can be said for other explicit expressions of time dependent behaviour in (2). In this paper, we have opted to retain the '$var(t)$' way of referring to the time dependent behaviour of pliant variables for readability. Conversely, any concrete syntax will need to contain notations for $t_{L(t)}$ and $t_{R(t)}$, since these may be needed to constrain intervals of time in the body of rel in a pliant rule.

As mentioned earlier, pliant variables can undergo instantaneous discontinuous transitions as well as continuous ones. For such transitions, the structure in (1) is sufficient.

We say that a continuous ASM ruleset is **well formed** iff:

- Every enabled mode transition is feasible, i.e. has an after-state, and on its completion enables a pliant transition (but does not enable any mode transition). (3)
- Every enabled pliant transition is feasible, i.e. has a time-indexed family of after-states, and EITHER: (4)

  (i) During the run of the pliant transition a mode transition becomes enabled. It preempts the pliant transition, defining its end. ORELSE
  (ii) During the run of the pliant transition it becomes infeasible: finite termination. ORELSE
  (iii) The pliant transition continues indefinitely: nontermination.

A **run** of a continuous ASM system starts with a mode transition which assigns the initial state of all system variables, and then, pliant transitions alternate with mode transitions. This reveals that the time points t_i introduced without comment earlier, are just the times at which the preemptions in (4) take place at runtime. The last transition (if there is one) is a pliant transition (whose duration may be finite or infinite).

The framework we have described so far is quite close to the modern formulation of hybrid systems as in [Tabuada 2009; Platzer 2010]. See also the large literature in the *Hybrid Systems: Computation and Control* series of international conferences, and the further literature cited therein.

4. CONTINUOUS ASM REFINEMENT AND RETRENCHMENT

Our approach to the development of complex systems is to master the complexity gradually, via top-down techniques. In this section we cover the refinement and retrenchment techniques we need.

4.1. The Discrete Case

In proving a conventional ASM refinement or retrenchment, we verify so-called (m,n) diagrams, in which m abstract steps simulate n concrete ones in an appropriate way. This means that there is nothing that the n concrete steps can do that is not suitably reflected in m appropriately chosen abstract steps, where both m and n can be freely chosen to suit the application. For this paper, it will be sufficient to focus on the refinement and retrenchment proof obligations (POs) which are the embodiment of this policy. The situation for refinement is illustrated in Fig. 2, in which we suppress input and output for clarity.

In Fig. 2 the retrieve relation $R_{A,C}$, between abstract and concrete states, holds at the beginning and end of the (m,n) pair. This permits us to 'glue together' such (m,n) diagrams to create relationships between abstract and concrete runs in which $R_{A,C}$ is periodically re-established. (N. B. In much of the ASM literature, the main focus is on an *equivalence*, usually written $\equiv$, between abstract and concrete states. This is normally deemed to contain a 'practically useful' subrelation $R_{A,C}$,

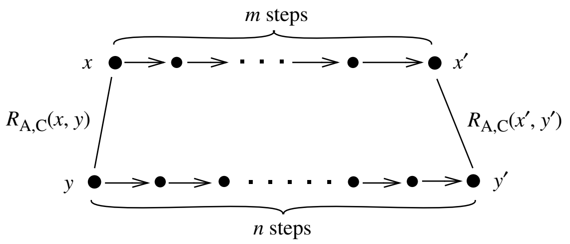

Fig. 2. An ASM (m,n) diagram, showing how m abstract steps, going from state x to state x' simulate n concrete steps, going from y to y'. The simulation is embodied in the retrieve relation $R_{\mathrm{A,C}}$, which holds for the before-states of the series of steps $R_{\mathrm{A,C}}(x,y)$, and is re-establshed for the after-states of the series $R_{\mathrm{A,C}}(x',y')$.

chosen to be easier to work with. The approach via $R_{\mathrm{A,C}}$ will be the focus of our treatment, and is also focus of the KIV [KIV] formalization in [Schellhorn 2001; 2005].)

The first PO is the initialization PO, and is common to both refinement and retrenchment:

$$\forall y' \bullet \mathit{CInit}(y') \Rightarrow (\exists x' \bullet \mathit{AInit}(x') \wedge R_{\mathrm{A,C}}(x',y')) \tag{5}$$

In (5), it is demanded that for each concrete initial state y', there is an abstract initial state x' such that the retrieve relation $R_{\mathrm{A,C}}(x',y')$ holds.

The second PO is correctness. The PO is concerned with the verification of (m,n) diagrams. For this, we have to have some way of deciding which (m,n) diagrams are sufficient for the application. Let us assume that we have done this. Let *CFrags* be the set of fragments of concrete runs that we have previously determined will permit a covering of all the concrete runs of interest for the application. We write $y::ys::y' \in \mathit{CFrags}$ to denote an element of *CFrags* starting with concrete state y, ending with concrete state y', and with intervening concrete state sequence ys. Likewise we write $x::xs::x' \in \mathit{AFrags}$ for abstract fragments. Let is, js, os, ps denote the sequences of abstract inputs, concrete inputs, abstract outputs, concrete outputs, respectively, belonging to $x::xs::x'$ and $y::ys::y'$ and let $\mathit{In}_{\mathrm{AOps,COps}}(is, js)$ and $\mathit{Out}_{\mathrm{AOps,COps}}(os, ps)$ denote suitable input and output relations. The specific form of the correctness PO now differs in form depending on whether we are dealing with refinement or retrenchment. We start with refinement. Then the correctness PO reads:

$$\begin{aligned} &\forall x, is, y, ys, y', js, ps \bullet y::ys::y' \in \mathit{CFrags} \wedge \\ &\quad R_{\mathrm{A,C}}(x,y) \wedge \mathit{In}_{\mathrm{AOps,COps}}(is, js) \wedge \mathrm{COps}(y::ys::y', js, ps) \Rightarrow \\ &\qquad (\exists xs, x', os \bullet x::xs::x' \in \mathit{AFrags} \wedge \mathrm{AOps}(x::xs::x', is, os) \wedge \\ &\qquad\quad R_{\mathrm{A,C}}(x',y') \wedge \mathit{Out}_{\mathrm{AOps,COps}}(os, ps)) \end{aligned} \tag{6}$$

In (6), it is demanded that whenever there is a concrete run fragment of the form $\mathrm{COps}(y::ys::y', js, ps)$, carried out by a sequence of concrete operations[3] COps, with state sequence $y::ys::y'$, input sequence js and output sequence ps, such that the retrieve and input relations $R_{\mathrm{A,C}}(x,y) \wedge \mathit{In}_{\mathrm{AOps,COps}}(is, js)$ hold between concrete and abstract before-states and inputs, then an abstract run fragment $\mathrm{AOps}(x::xs::x', is, os)$ can be found to re-establish the retrieve and output relations $R_{\mathrm{A,C}}(x',y') \wedge \mathit{Out}_{\mathrm{AOps,COps}}(os, ps)$.

The ASM refinement policy also demands that non-termination be preserved from concrete to abstract. We will not need that though, in the case study of this paper.

Assuming that (5) holds, and that we can prove enough instances of (6) to cater for the application of interest, then the concrete model is a **correct refinement** of the abstract model. In a correct refinement, all the properties of the concrete model (that are visible through the retrieve and other

[3]We define an operation as a maximal enabled set of rules — provided its updates are consistent. Enabled inconsistent updates cause abortion of the run, as usual in ASM.

relations), are suitably reflected in properties of the abstract model (because of the direction of the implication in (6)). If in addition, the abstract model is also a correct refinement of the concrete model (using the converses of the same relations), then the concrete model is a **complete refinement** of the abstract model. In a complete refinement, all relevant properties of the abstract model are also present in the concrete model (because of the direction of the implication in the modified version of (6)). Therefore, to ensure that the complete set of requirements of an intended system is faithfully preserved through a series of refinement steps, it is enough to express them all in a single abstract model, and then to ensure that each refinement step is a complete refinement. We now turn to the retrenchment version of the correctness PO.

For retrenchment, [Banach et al. 2007; Banach et al. 2008] give definitive accounts; latest developments are found in [RET]. The retrenchment correctness PO weakens (6) by inserting *within*, *output* and *concedes* relations, $W_{\text{AOps,COps}}$, $O_{\text{AOps,COps}}$, $C_{\text{AOps,COps}}$ respectively into (6), to give extra flexibility and expressivity. In particular, the concession $C_{\text{AOps,COps}}$ weakens the conclusions of (6) disjunctively, giving room for many kinds of 'exceptional' behaviour. The result is:

$$
\begin{aligned}
&\forall x, is, y, ys, y', js, ps \bullet y :: ys :: y' \in CFrags \wedge \\
&\quad R_{\text{A,C}}(x,y) \wedge W_{\text{AOps,COps}}(is, js, x, y) \wedge \text{COps}(y :: ys :: y', js, ps) \;\Rightarrow \\
&\quad\quad (\exists xs, x', os \bullet x :: xs :: x' \in AFrags \wedge \text{AOps}(x :: xs :: x', is, os) \wedge \\
&\quad\quad\quad ((R_{\text{A,C}}(x', y') \wedge O_{\text{AOps,COps}}(x, x', is, os, y, y', js, ps)) \vee \\
&\quad\quad\quad C_{\text{AOps,COps}}(x, x', is, os, y, y', js, ps)))
\end{aligned} \tag{7}
$$

To ensure that retrenchment only deals with well defined transitions, and to ensure smooth retrenchment/refinement interworking, in retrenchment we also insist that the values at which $R_{\text{A,C}} \wedge W_{\text{AOps,COps}}$ holds always falls inside the domain of the requisite operations, though this is another thing that will not be needed in our case study.

4.2. The Continuous Case

All of the preceding was still formulated for the exclusively discrete world. However, to extend it to the continuous world too, is not very hard. The essence of the approach is to reinterpret the run fragments $AOps(x::xs::x', is, os)$ and $COps(y::ys::y', js, ps)$ appearing in (6) and (7) appropriately.

In the discrete context, these notations refer to a sequence of states, i.e. a map from some natural number indexes to states. In the continuous context, the analogue of this is a map from an interval of time to states. Given our framework of alternating mode transitions and pliant transitions, the abstract $x::xs::x'$ and concrete $y::ys::y'$ should each consist of an alternating sequence of mode and pliant transitions (starting with either kind and ending with either kind).

We thereby derive an interpretation for the notations used in (6) and (7) appropriate for the current context. Thus $R(x,y)$ becomes a predicate about the earliest abstract and concrete state values referred to by the state functions mentioned, similarly $R(x',y')$ refers to the latest state values. Predicates such as $O_{AOps,COps}(x, x', is, os, y, y', js, ps)$ refer to these earliest and latest state values, and to the whole of the input and output functions as appropriate. In this way, (6) and (7) continue to define ASM refinement and retrenchment in the enlarged context. The piecing together of (m,n) diagrams to build an abstract simulation of a concrete run, now reduces to the identification of the latest (abstract and concrete) state values reached by one (m,n) diagram, with the earliest (abstract and concrete) state values of its successor (m,n) diagram.

In the light of the above remarks, it is instructive to point out what the POs (6) and (7) *do not* stipulate. We have already said that the ASM POs do not mention states that are internal to the fragments that occur in a given (m,n) diagram. This frequently simplifies the relations that capture the relationship between abstract and concrete worlds. Also, it is not demanded that the abstract and concrete run fragments in one of our (m,n) diagrams *both* have to start with a transition of the same kind (i.e both mode or both pliant) — neither do they both have to end with a transition of the same kind. Finally, there is no explicit mention of time in (6) and (7). In particular there is nothing in either (6) or (7) that indicates how time is expected to progress during the respective length m and length n fragments. The designer of the models is free to design the progress of time in the abstract and concrete systems in the way most appropriate as judged from the wider requirements arena, and any design must stand up to scrutiny from this perspective.

4.3. Large Scale Development, Time Dependent Retrieve Relations, and the Tower Pattern

In large scale formal development, despite it being the ideally desired practice that all development steps are performed using (some chosen notion of) refinement, real world requirements issues often prevent formal modeling in the most direct and transparent way. It might be necessary to bring the problem description within the capabilities of the formal modeling framework, or to bring the complexity of the description down to a more tractable level. It might be necessary to address some entirely non-technical issues, such a the perspicuity of the model to non-technical stakeholders, or a desired conformance to pre-existing systems, which are not strictly compatible with the system being currently developed.

Retrenchment was introduced [Banach et al. 2007; Banach et al. 2008] in order to provide more flexibility to the formal development process, at the price of a decrease in the level of the formal guarantees that its use can offer. Obviously, it is preferable to use refinement as much as possible, and to use retrenchment as little as possible. To integrate the two techniques into a coherent whole, compatibility results are needed, and these are thoroughly investigated in [Banach and Jeske 2014]. The results in [Banach and Jeske 2014] are known as the Tower Pattern, and this is illustrated in the left hand side of Fig. 3, which shows a typical ‘commuting square’, with refinements vertical and retrenchments horizontal. The results of [Banach and Jeske 2014] show that given two adjacent sides of such a square, the remainder can be reconstructed in a canonical way. This leads to visualising a development process that utilises both techniques in the way we show on the right of Fig. 3, in which a refinement+retrenchment development path zig-zags its way through a grid of system models. This takes downwards directed refinement steps when the requirements issues handled are compatible enough with the preceding system for refinement, and takes horizontally directed retrenchment steps when they are not.

In this paper, we want to use the development pattern just described, to address a specific tricky issue that appears in the development of systems whose abstract models are most naturally framed in continuous/hybrid terms, yet where the implementation target platform is digital, and so the control to be exerted must ultimately be framed in discrete terms alone.

The tricky issue we are interested in here, which arises in passing from the continuous/hybrid to the purely discrete world, is the quality of the approximation involved in doing so. Ideally, any discrepancy between continuous and discretized states should always remain within a statically determined bound. This permits a statically determined retrieve relation of the form $R_{\mathrm{A,C}}(x,y) \equiv (|y - f(x)| \leq \delta)$ to be maintained, where x and y are the abstract and concrete states, and f is a ‘natural’ correspondence between them (most commonly an identity). Although stable control systems satisfy this property, there are many other kinds of system which are not so benign.

In the more general case, the quantity which we called $R_{\mathrm{A,C}}$ above, becomes a more dynamic quantity, depending on time and on the detailed abstract and concrete dynamical trajectories. The discrepancy between a continuous trajectory and a given discretization of it may grow and shrink over time, precluding the use of a static $R_{\mathrm{A,C}}$ as used in rules like (6), where it appears in both hypothesis and conclusion.

In the absence of a static $R_{\mathrm{A,C}}$ in the dynamic case, it is tempting to continue to work with its dynamic counterpart, a time dependent $R_{\mathrm{A,C}}$. However, this silently nullifies a premiss normally assumed in doing refinement, namely that quantities used for describing the refinement, such as $R_{\mathrm{A,C}}$, are supposed to be *invariants*, i.e. quantities that do not change over time.

The Tower Pattern of refinements and retrenchments described above offers, we believe, a better account of this type of dynamic situation. It gives us the option of using the within relation of a retrenchment to house facts concerning the discrepancy between the continuous and discretized states at the begining of a pliant transition, and the output relation to house the corresponding (but

$$
\begin{tikzcd}
\mathsf{A} \arrow[r, "Ret_{\mathsf{A,B}}"] \arrow[d, "Ref_{\mathsf{A,C}}"'] & \mathsf{B} \arrow[d, "Ref_{\mathsf{B,D}}"] \\
\mathsf{C} \arrow[r, "Ret_{\mathsf{C,D}}"'] & \mathsf{D}
\end{tikzcd}
$$

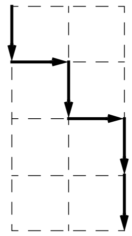

Fig. 3. On the left, the basic refinement/retrenchment commuting square, with the corners occupied by system models A, B, C, D, where the systems are pairwise related by refinements which are vertical arrows and retrenchments which are horizontal arrows. On the right, a schematic development, shown heavy, consisting of refinements and retrenchments — and the grid of systems it is embedded in by virtue of refinement/retrenchment square completions.

quantitatively different) facts concerning the discrepancy at its end. This is sufficient for 'refinement-like' situations, where the behaviour of continuous and discretized systems remains broadly similar. Beyond that, we also have the concedes relation, allowing us to accommodate situations in which the discretization process introduces wilder deviations between the abstract and concrete systems.

In the situations described, a generic and useful retrieve relation is given by true. It leaves the whole job of controlling the abstract/concrete discrepancy to the within and output relations.

In the case study below, we see an example of the use of this development structure in our treatment of discretization. We also see retrenchment used for a less conventional purpose, that of relating a reference model for a normal heartbeat (in which the heartbeat is an output), to a sensing model (in which the heartbeat is an input), the latter of which may furthermore observe heart behaviour incompatible with the expected behaviour of a normal heart.

The distinction between a genuinely static retrieve relation $R_{\mathrm{A,C}}$ and a time dependent 'pseudo-$R_{\mathrm{A,C}}$' can be appreciated as an interchange of (meta level) quantifiers. A static retrieve relation in effect says '*there exists* a retrieve relation $R_{\mathrm{A,C}}$ (depending only on the state spaces) such that *for all* suitably matched pairs of simulating steps, run fragments etc., $R_{\mathrm{A,C}}$ is preserved by the simulation'. By contrast, a dynamic retrieve relation interchanges these quantifiers, saying '*for all* suitably matched pairs of simulating steps, run fragments etc., *there exists* a retrieve relation $R_{\mathrm{A,C}}$ (which must now be allowed to depend on the simulating pair in question and on time) such that $R_{\mathrm{A,C}}$ is preserved by the simulation'. Particularly when we have a top-to-bottom development process, including a range of diferent types of development step, keeping a clear distinction between static retrieve relations on the one hand, and dynamic retrieve relations, expressed using the within and output or concedes relations of a retrenchment, is conceptually clarifying in our opinion.

It is clear enough (and is discussed in detail in [Banach et al. 2008]) that the composition of any sequence of retrenchments and refinements containing at least one retrenchment, is itself a retrenchment. Analogously, the picture based on meta level quantification makes it clear that the composition of any sequence of retrieve relations at least one of which is time dependent, is itself time dependent. So treating the time dependent ones via retrenchment turns out to be appropriate.

5. PACEMAKER SENSING

Our objective for the rest of this paper is to design an ASM system to track a signal that is intended to represent the continuous internal electrical activity of the heart, and to decide whether it conforms to the expected shape of a spontaneously generated depolarization. Starting from the end of the last heartbeat, if, within a specified maximum period, the received signal is judged to contain a pulse of the right shape, the heart is judged to have behaved normally and the pacemaker is prevented from issuing a pulse of its own. On the other hand, if no such pulse is detected, the heart is judged to be abnormal, and the pacemaker will issue a pulse to maintain appropriate circulation of the blood.

A theoretical treatment of the expected shape of the signal detected by an electrode inside the heart as a wave of depolarization passes over it has been carried out some time ago, based on the

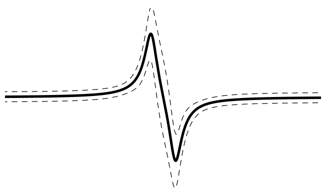

Fig. 4. Schematic of a typical normal internal cardiac electrogram corresponding to a single depolarization, with a margin of error to allow for noise.

idea of a dipole of charge passing a detector [Wilson et al. 1933a; 1933b]. The cited references consider more sophisticated embellishments of the simple dipole idea, for instance due to a dipole wavefront passing through a sheet of excitable tissue (such as the myocardium), but the essential shape of the pulse changes very little. More significant are distortions due to proximity of the sensing electrode to the source of the depolarization, and other biological factors. For our purposes though, the basic shape in Fig. 1.(b) will do.

Thus, our abstract model focuses on the shape in Fig. 1.(b), allowing for an acceptable margin of deviation during normal activity. This is shown in Fig. 4. There, the basic shape $pulse(t)$ (solid curve in Fig. 4) is given by a formula of the form:

$$pulse(t) \ = \ -K\left(\frac{1}{\sqrt{(t+a)^2+b^2}}-\frac{1}{\sqrt{(t-a)^2+b^2}}\right) \tag{8}$$

with K positive, while the dashed error curves surrounding it are given by adding (or subtracting) $err(t)$, which is a small positive constant augmented by a strongly peaked Gaussian centred at the point of maximum variability of $pulse(t)$ (to allow greater leeway in the region of maximum change of the ideal signal):

$$err(t) \ = \ c + r\exp(-pt^2) \tag{9}$$

In real pacemakers, the completion of the previous heartbeat initiates the start of a fresh cycle. Let us assume that this is at time $t = 0$. The new cycle begins with a so called refractory period, in which the heart signal is effectively ignored, allowing the transients of the previous heartbeat to die off, and for myocardial polarization to be re-established. (The latter can be detected as the ‘T wave’ in Fig. 1.(b).) Say this refractory period lasts till T_0. Then, from T_0 to T_{MAX}, the heart signal is monitored for the presence of a shape similar to *pulse*. Assuming the heart is working normally, if the patient is being physically active, then the patient’s heartrate will be higher, and a *pulse*-like shape occurs sooner; whereas if the patient is being inactive then the heartrate will be lower, and a *pulse*-like shape will occur later, although always within the given window. Thus, we can specify a typical function $egm_A(t)$ that constitutes an acceptable electrogram shape as follows:

$$\begin{aligned}&(\exists R \bullet T_0 < R_0 \leq R \leq R_{MAX} < T_{MAX} \wedge \\ &\quad (\forall t \bullet T_0 < t < T_{MAX} \Rightarrow egm_A(t) \in egmWin_R(t)))\end{aligned} \tag{10}$$

where

$$egmWin_R(t) = [egmWin_R^-(t) \ldots egmWin_R^+(t)] \tag{11}$$

and

$$egmWin_R^{\pm}(t) = pulse(t-R) \pm err(t-R) \tag{12}$$

In (10), $egmWin_R(t)$ is set valued, and at each relevant time t, it specifies the allowable cardiac electrogram window within which a normally operating heart’s electrogram behaviour is expected to fall. A minor complication is that the precise moment within the allowed timeframe at which the electrogram undergoes the pulse is not known exactly, but is merely constrained by $R_0 \leq R \leq R_{MAX}$, where, like T_0 and T_{MAX}, the constants R_0 and R_{MAX} are statically determined.

6. ASM SENSING MODELS

In this section, based upon the preceding remarks, we present a succession of ASM models that take us from the most abstract formalisation of the account just given, to a model at a level close to implementation. Along the way we encounter a multitude of technical issues that can impact the development (in the discrete domain) of a non-trivial continuous application.

6.1. The Continuous ASM Heartbeat Reference Model

At the most abstract level, we can view $egmWin_R(t)$ as a static function, depending on two parameters: time and the chosen value of R. With this, we can specify a pliant rule that generates acceptable electrograms as follows, where any call of EGM_A produces an electrogram egm_A which stays within the acceptable window $egmWin_R$ during the period of interest, $T_0 < t < T_{MAX}$:

$$\begin{array}{l} \text{EGM}_A(\textbf{out}\ ego(t)) \ \stackrel{\text{c}}{=} \\ \textbf{choose}\ \ eg'(t)\ \ \textbf{with}\ \ (\exists R \bullet T_0 < R_0 \leq R \leq R_{MAX} < T_{MAX}\ \wedge \\ \qquad\qquad\qquad\qquad\quad T_0 < t < T_{MAX} \wedge eg'(t) \in egmWin_R(t)) \\ \quad \textbf{do}\ \ egm(t)\ := \ eg'(t), \\ \qquad\ \ ego(t)\ := \ eg'(t) \end{array} \tag{13}$$

Eq. (13) is an instance of the direct assignment version of a pliant rule.

One technical point connected with (13) is that since $egmWin_R$ is set-valued, egm and ego, as specified by EGM_A, are merely constrained to lie within the prescribed set for every relevant t, permitting EGM_A to generate (mathematically) highly pathological functions in addition to well-behaved ones. In an *implementation* context, there is no obstacle to prevent a design process from restricting to well-behaved functions only. Likewise, in the present context, a biologically based *observation* context, the observed heart behaviour will be mathematically well-behaved too, simply because biological processes cannot move fast enough to generate (mathematically) pathological behaviours. Such properties of the external environment naturally belong in the domain theory that provides the context for the system being formally developed. We return to this point below.

We can regard EGM_A as a reference model for cardiac behaviour (outside of the refractory period), serving as a guide for more detailed modeling, next.

6.2. Concrete ASM Sensing

The previous model failed to take into account a number of important points. Firstly is that we are ultimately interested in *sensing* rather than *generating* heartbeats. So the outputs we generated from EGM_A have to be related to inputs in a more realistic setting. Secondly is that heartbeats are not all of constant duration, i.e. they do not all last from T_0 till T_{MAX}. As soon as a pulse has occurred, the repolarization process starts, and the next heartbeat is under way — it is easy enough to deal with this by adjusting the constraints in the model. Thirdly, various technical issues crop up in the detailed definition of what sensing actually means in the continuous sphere which have to be addressed. Finally, in practice, the pacemaker does not monitor the electrogram continuously, but at evenly spaced discrete moments. The issue then arises of connecting this discrete practical mode of working with the ideal continuous observation of the continuous behaviour from which it emerges. We tackle these aspects in stages.

6.3. Heartrate-Aware ASM Sensing

We now present the continuous variable heartrate sensing model, the *CVH* model. Initialisation (assumed to be at time 0), resets the 'verdict' variables *pulseGOOD* and *pulseBAD*, and initialises the pliant variables *winR*, *dev*, *egirise* and *egifall* which will measure the quality of the electrogram:

$$\begin{array}{l}\textsc{Init}_{CVH} = \\ \textbf{if}\ \ t = 0\ \ \textbf{then} \\ \quad \textbf{do}\ \ mode\ := \ refractory, \\ \quad\quad pulseGOOD\ := \ \mathsf{false},\ pulseBAD\ := \ \mathsf{false}, \\ \quad\quad winR\ := \ \emptyset,\ dev\ := \ 0, \\ \quad\quad egirise\ := \ 0,\ egifall\ := \ 0 \end{array} \tag{14}$$

The model of the refractory period simply skips:

$$\textsc{Refractory}_{CVH} \ \overset{\mathrm{c}}{=}\ \ \textbf{if}\ \ mode = refractory\ \ \textbf{then do}\ \ \mathsf{skip} \tag{15}$$

A mode transition signals the start of sensing. Some of the work of initialisation is repeated as this is the re-entry point of sensing for subsequent heartbeats:

$$\begin{array}{l}\textsc{StartSensing}_{CVH} = \\ \textbf{if}\ \ mode = refractory \wedge t = T_0\ \ \textbf{then} \\ \quad \textbf{do}\ \ mode\ := \ sensing, \\ \quad\quad pulseGOOD\ := \ \mathsf{false},\ pulseBAD\ := \ \mathsf{false}, \\ \quad\quad winR\ := \ \emptyset,\ dev\ := \ 0, \\ \quad\quad egirise\ := \ 0,\ egifall\ := \ 0 \end{array} \tag{16}$$

The switch to sensing mode activates the sensing process. Sensing continues until a signal conforming to a number of the characteristics of the region of high variation in *pulse* has been detected (indicating a spontaneous heartbeat), or the timeout expires (indicating abnormal cardiac function). Sensing can be described using pliant rule (17).

In (17), Θ returns 1 if its parameter is positive, else it returns 0. Thus, the derivative of *egirise* equals the derivative of *egi* provided the latter is positive, and consequently *egirise* mirrors *egi* as it is increasing, remaining stationary during periods when *egi* is decreasing. By this means, *egirise* accumulates the total increase of *egi* during increasing episodes. Analogously, *egifall* accumulates the total decrease of *egi* during decreasing episodes.

For the rest of (17), let us take the case of a normal heartbeat to start with. We assume that on exiting the refractory period, at T_0, the sensed electrogram signal (i.e. the input $egi(T_0)$) is very near zero. If $egi(t)$ is close to zero near $t = T_0$, then a large range of R values (basically, any acceptable value $R_0 \le R \le R_{MAX}$) will satisfy $egi(t) \in egmWin_R(t)$ — because to the left of the double spike, $egmWin_R$ is just a sausage enclosing the time axis (see Fig. 4). In this regime therefore, $wR(t)$, and thus also $winR(t)$, will be large sets.

As time progresses beyond a little less than R_0, assuming the double spike does not occur yet, the lower bound of $winR(t)$ (which monitors the lowest possible R value that could describe the heartbeat), increases smoothly to avoid contradicting the observed electrogram.

$$\begin{array}{l}\textsc{Egm}_{CVH}(\textbf{in}\ egi(t))\ \overset{\mathrm{c}}{=} \\ \textbf{let}\ \ wR(\bar{s}) = \boldsymbol{if}\ \bar{s} \notin \mathrm{dom}\ egi\ \boldsymbol{then}\ \emptyset \\ \qquad\qquad\quad \boldsymbol{else}\ \{R \mid R_0 \le R \le R_{MAX} \wedge \\ \qquad\qquad\qquad\qquad \forall \tilde{s} \bullet t_{\mathrm{L}(t)} < \tilde{s} \le \bar{s} \Rightarrow egi(\tilde{s}) \in egmWin_R(\tilde{s})\}\ \boldsymbol{fi}\ \ \textbf{in} \\ \textbf{let}\ \ s = \max\{\tilde{s} \mid t_{\mathrm{L}(t)} < \tilde{s} \le t \wedge wR(\tilde{s}) \ne \emptyset\}\ \ \textbf{in} \\ \textbf{if}\ \ mode = sensing\ \ \textbf{then} \\ \quad \textbf{do}\ \ winR(t)\ := \ wR(s), \\ \quad\quad dev(t)\ := \ \textbf{solve}\ \ \mathcal{D}\,dev(t) = \min\{\mathrm{dist}(egi(t), egmWin_R(t)) \mid R \in winR(t)\}, \\ \quad\quad egirise(t)\ := \ \textbf{solve}\ \ \mathcal{D}\,egirise(t) = \mathcal{D}\,egi(t)\,\Theta(\mathcal{D}\,egi(t)), \\ \quad\quad egifall(t)\ := \ \textbf{solve}\ \ \mathcal{D}\,egifall(t) = \mathcal{D}\,egi(t)\,\Theta(-\mathcal{D}\,egi(t)) \end{array} \tag{17}$$

As time progresses further, the normal heartbeat reaches the point at which the double spike occurs. As soon as $egi(t)$ increases beyond the upper bound of the 'left hand sausage' part of *pulse*, the upper range of R values for which the whole signal sits within $egmWin_R$ decreases dramatically. As the electrogram follows the shape of *pulse*, very soon the range of allowable R values is reduced to approximately $2t_0$ (where t_0 is the solution nearest to 0 to the equation $pulse(t_0) + err(t_0) = 0$), which is the width of the time window around the central point of *pulse* in Fig. 4 — the right hand part of Fig. 7 shows the overall behaviour. Setting δ_{th} (the maximum permitted width of the window of allowable R values such that $egmWin_R(t)$ (for all the relevant values of t) contains the whole of the observed electrogram) to slightly more than $2t_0$, gives us a feature to test for when confirming the presence of a normal heartbeat. We also note that for a normal heartbeat, in practice, the width of the window of allowable R values will be nonzero, since there will always be some leeway surrounding the fit of a realistic signal within the permitted envelope, yielding another feature to test for when confirming a normal heartbeat.

Beyond that, since the whole of the observed electrogram is inside $egmWin_R$, according to (17), $\mathcal{D}\,dev(t)$, the derivative of the variable *dev*, remains at zero, since the distance between $egi(t)$ and $egmWin_R(t)$ (for any acceptable value of R) is always zero. So *dev* stays at its initial value, namely zero. This gives us another feature to test for when confirming a normal heartbeat. Furthermore, since, as discussed above, *egirise* and *egifall* track the overall positive and negative changes in the electrogram value, a normal heartbeat will also be characterised by $|egirise + egifall| < \delta_{rf}$ and $egirise > \Delta_{th}$, with δ_{rf} and Δ_{th} defined as suitable constants, the former small, the latter big. Finally, observing also that the time for confirming a normal heartbeat will be less than the timeout T_{MAX}, we see that a normal heartbeat enables the following mode transition, which will preempt further sensing:

$$
\begin{array}{l}
\text{PULSEGOOD}_{CVH} = \\
\textbf{if } mode = sensing \wedge T_0 < t < T_{MAX} \wedge dev = 0 \wedge \\
\quad 0 < |\max winR - \min winR| < \delta_{th} \wedge |egirise + egifall| < \delta_{rf} \wedge \\
\quad egirise > \Delta_{th} \;\; \textbf{then} \\
\quad\; \textbf{do } mode := refractory, \\
\qquad\;\; pulseGOOD := \mathsf{true}
\end{array} \tag{18}
$$

We turn to the case of an abnormal heartbeat. Again, close to $t = T_0$, the electrogram will start near zero. Therefore, close to $t = T_0$, the situation will be as before, with $wR(t)$ and $winR(t)$ large. However, by definition, an abnormal heartbeat does *not* conform to the envelope of permitted deviation around *pulse*, given by $egmWin_R$, for any $R_0 \leq R \leq R_{MAX}$. Thus the originally large sets $wR(t)$ and $winR(t)$ will, in time, diminish, but instead of eventually remaining at a size a little less than δ_{th}, the two sets will ultimately shrink to a single value, the last value $\bar{R}$ at the last time $\bar{t}$ for which the property 'the electrogram-so-far is contained within $egmWin_R$ for at least one permitted value of R' still holds (before it fails for the remainder of the heartbeat).

After this last moment $\bar{t}$, the distance between $egi(t)$ and $egmWin_{\bar{R}}(\bar{t})$ will become positive, leading to a positive derivative $\mathcal{D}\,dev(t)$. When integrated, this leads to a positive contribution to *dev* itself, and since the RHS of the DE is a distance, which is always non-negative, *dev* will be positive when the heartbeat deadline T_{MAX} expires, giving us a feature to test for when confirming the presence of an abnormal heartbeat. We see that an abnormal heartbeat enables the following mode transition, which will preempt further sensing:

$$
\begin{array}{l}
\text{PULSEBAD}_{CVH} = \\
\textbf{if } mode = sensing \wedge t = T_{MAX} \wedge dev > 0 \;\; \textbf{then} \\
\quad\; \textbf{do } mode := refractory, \\
\qquad\;\; pulseBAD := \mathsf{true}
\end{array} \tag{19}
$$

We see that the sensing of an individual heartbeat lasts just long enough to detect a spontaneous depolarisation in the normal case, or lasts till the deadline expires in the abnormal case.

To deal with successive heartbeats, we need to slightly alter the way we deal with time. There are two relatively straightforward approaches. In the first, we continue with the perspective that t refers to real time (measured from some arbitrary starting point). In this view, we would need PULSEGOOD_{CVH} and PULSEBAD_{CVH} to re-assign the values of T_0 and T_{MAX} (and therefore also R_0 and R_{MAX}) to appropriate future values (by adding to them the duration of the heartbeat that has just elapsed), ready for the next heartbeat. In the second, we would regard t as a *clock* —which is reset at

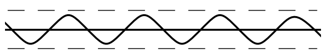

Fig. 5. A hypothetical sawtooth-like electrogram signal, leading to an accumulation of positive and negative episodes, each potentially aggregating to a size Δ_{th}, but without a significant departure from the zero signal.

the beginning of every heartbeat (via assignments $t := 0$ in PULSEGOOD$_{CVH}$ and PULSEBAD$_{CVH}$)—instead of referring to real time. In this view, we would need to re-interpret the notations $t_{\mathrm{L}(t)}$ and $t_{\mathrm{R}(t)}$ so that they referred to the clock time at the beginning and end of the current invocation of the current pliant transition, rather than to real time. Both of these approaches are quite straightforward to write down in detail, allowing REFRACTORY$_{CVH}$ to be re-enabled for a repetition of the sensing behaviour in the next heartbeat.

Having completed the description of the model, we pause to discuss some of its less obvious aspects. Firstly, we note that the definition of EGM$_{CVH}$ is *exact*, in that we can argue that the property $dev = 0$ is equivalent to the heart's behaviour being consistent with a natural heartbeat. Therefore, one way of distinguishing good from bad heartbeats, is to wait till the deadline T_{MAX}, and then check if that value of dev is non-zero. However, this takes us back to the 'what exactly are the requirements?' question posed at the beginning of the paper. If the patient is being active, and a natural heartbeat has occurred, we want to be getting on with the next heartbeat as soon as one has finished. Therefore, we need to design mode rules that are able to react in a timely manner, which is the purpose of PULSEGOOD$_{CVH}$ and PULSEBAD$_{CVH}$.

The verdict variables, *pulseGOOD* or *pulseBAD*, encapsulate a binary decision (or none if neither fires — we do not consider the borderline case in which both fire simultaneously). On the other hand, the data they are based on is continuous, the value of dev. It is thus mathematically possible for dev to become arbitrarily small but non-zero, and the criteria for good and bad behaviour to be arbitrarily difficult to distinguish. We have thus elicited an additional requirements issue, namely that the clinical reality is such that there will generally be a clear enough demarcation line between good and bad heartbeats, so that they can be distinguished by the above modeling provided that the various constants are set appropriately. We assume that this will be the case if the sensing we are describing is also supplemented by the frequency domain sensing outlined in Section 2.

There is thus no escaping the fact that the design of the mode rules PULSEGOOD$_{CVH}$ and PULSEBAD$_{CVH}$ *defines* what is and what is not an acceptable electrogram signal, and that this will allow through pathological cases of one kind or another. As an example, consider a candidate electrogram signal as follows: it starts near zero at T_0 but has a 'sawtooth-like' behaviour, in which the teeth of the sawtooth are small and rounded. Since the teeth are small, the signal stays within the bounds of the 'left hand sausage' part of Fig. 4. But the fact that the signal moves up and down a lot means that the accumulated positive and negative changes in the electrogram value may exceed the Δ_{th} threshold. See Fig. 5. So the PULSEGOOD$_{CVH}$ rule could fire even though there is no actual double spike in the signal. More sophisticated designs for PULSEGOOD$_{CVH}$ and PULSEBAD$_{CVH}$ can alleviate the risks one or other pathology at the price of admitting another, and the justification for accepting one design or another always rests in the application domain.

6.4. On EGM$_A$ and EGM$_{CVH}$

An interesting question concerns the formal relationship between the abstract EGM$_A$ electrogram function and the more concrete EGM$_{CVH}$ sensing model. In one sense, there is no functional connection between them at all. After all, one is a model of aspects of the heart, and the other a device that monitors and stimulates the heart — one can hardly imagine the device to be a refinement of the organism.

And yet, exploring a formal connection between them is potentially helpful. In being able to precisely state the connection between them, we clarify the extent to which system requirements have or have not been adequately addressed in the system model — in this paper we comment often on how troublesome the requirements issue is in our case study.

Accepting then that establishing a formal relationship between the EGM_A and EGM_{CVH} models has some value, the most desirable such relationship between them is of course a refinement, but such an idea fails in this case for a number of reasons.

First of all, the main use of pacemakers is in situations where cardiac behaviour is expected to *not* conform to the normal ideal for a significant proportion of the time, making the prospects for any conventional kind of refinement futile. Secondly, even disregarding the cases of abnormal cardiac behaviour, we note that since the width of the allowable window $egmWin_R$ varies over time, and sensed heartbeats are in general not of the same duration as EGM_A, it would not be possible to define a retrieve relation R that depended purely on state values, and yet catered for the variability we expect to see.

On the other hand, if we take advantage of the additional flexibility offered by the more expressive retrenchment framework, then the prospects for setting up a formal relationship between the two models are significantly improved. We give one possible such retrenchment from EGM_A to EGM_{CVH} now. Observe that we suppress irrelevant variables below (principally, the mode variables).

The first concern in a retrenchment is the retrieve relation from the A model state space to the CVH model state space, $R_{A,CVH}$. We just argued that it would be difficult to find a convincing $R_{A,CVH}$ that depended purely on state values. Under such circumstances, in line with our discussion in Section 4.3, and recognising that the other relations of a retrenchment permit more finegrained control over states, it is acceptable to trivialise $R_{A,CVH}$:

$$R_{A,CVH}(egm, \langle winR, dev \rangle) \equiv \mathsf{true} \tag{20}$$

The within relation controls what is demanded of before-states and inputs in the hypotheses of the correctness PO (7):

$$\begin{aligned} & W_{\mathrm{EGM}_A,\mathrm{EGM}_{CVH}}(egi(t \in (T_0 \dots T')), egm(T_0), \langle winR, dev, egirise, egifall \rangle (T_0)) \equiv \\ & \quad dev(T_0) = 0 \wedge egirise(T_0) = 0 \wedge egifall(T_0) = 0 \wedge |egi(T_0)| < \delta_{\mathrm{small}} \end{aligned} \tag{21}$$

In (21), T' refers to the time at which the after-state of the heartbeat is reached. We see that $dev(T_0) = 0$ and $egirise(T_0) = 0$ and $egifall(T_0) = 0$ are all recorded in (21), consistent with (16), and despite the fact that these initial values are not mentioned in (17). We also record that the magnitude of the right limit of the sensed electrogram, $egi(T_0)$, is small enough to substantiate our earlier assumptions that egi is near zero at the start of sensing.

Regarding the consequences of the execution of EGM_A and EGM_{CVH}, either EGM_{CVH} conforms to EGM_A and we have a normal heartbeat, or not. In the former case, it is appropriate to describe the properties of the two executions using the output relation:

$$\begin{aligned} & O_{\mathrm{EGM}_A,\mathrm{EGM}_{CVH}}(egm(T_0, T'), ego(t \in (T_0 \dots T')), \\ & \qquad\qquad \langle winR, dev, egirise, egifall \rangle (T_0, T'), egi(t \in (T_0 \dots T'))) \equiv \\ & \quad dev(T') = 0 \wedge 0 < |\max winR - \min winR| < \delta_{th} \wedge \\ & \quad |egirise + egifall| < \delta_{rf} \wedge egirise > \Delta_{th} \wedge \\ & \quad (\forall t \bullet T_0 < t < T' \Rightarrow \mathrm{EGM}_A(ego(t)) \wedge ego(t) = egi(t)) \end{aligned} \tag{22}$$

The gist of (22) is that the variable dev is zero in the after-state, which, from our earlier discussion, implies that it must have remained at zero throughout the pliant transition. Besides that, the accumulated rise in the observed electrogram signal was big enough to witness a genuine double spike, and the signal returned to near zero since the sum of the accumulated rise and the accumulated fall is small, i.e. less than δ_{rf}. (These latter observations are made on the assumption that the possibility of mathematically pathological origins for the values of *egirise* and *egifall* could be disregarded, e.g. the sawtooth behaviour discussed earlier.) Also, since we had a normal heartbeat, the input electrogram $egi(t)$ always remained within the permitted envelope, $egmWin_R(t)$ for some R, and therefore there is a possible abstract electrogram $ego(t)$ that followed it exactly. This is recorded in the last clause of (22).

If though, we are dealing with the case of an abnormal heartbeat, we can describe the properties of the two executions using the concession:

$$
\begin{aligned}
& C_{\mathrm{EGM}_A,\mathrm{EGM}_{CVH}}(egm(T_0,T'), ego(t \in (T_0 \ldots T')), \\
& \qquad\qquad \langle winR, dev, egirise, egifall\rangle(T_0,T'), egi(t \in (T_0 \ldots T'))) \; \equiv \\
& \quad T' = T_{MAX} \wedge dev(T') > 0 \wedge \\
& \quad (\forall R \bullet R_0 \leq R \leq R_{MAX} \Rightarrow (\exists t \bullet T_0 < t < T' \wedge egi(t) \notin egmWin_R(t)))
\end{aligned} \tag{23}
$$

In (23), the first clause states that the deadline has expired, while the second states that the accumulated deviation from (even the best possible) permitted electrogram windows is positive. The third clause makes the preceding more explicit by asserting the existence of a value t at which the observed electrogram falls outside the permitted window $egmWin_R(t)$ (for even the best R).

An interesting question that arises in connection with the above, is that of *proving* the retrenchment verification conditions built using the presented relations. The fact that we were able to point to potential ways of spuriously enabling the PULSEGOOD$_{CVH}$ rule via a mathematically pathological counterexample (the sawtooth behaviour discussed above), indicates that the path from basic assumptions to discharge of the relevant POs need not be completely straightforward. For instance, without rigorous proof, we cannot exclude the possibility of there being yet other kinds of counterexample that make the rules we have written behave in a way that we would not want. And the ease or difficulty of achieving such proofs depends strongly on what we put into the *domain theory* that underpins our models. A stronger domain theory facilitates proof, while a weaker domain theory inhibits proof. Moreover, how strong we make our domain theory connects strongly with the ‘what exactly are the requirements?’ question mentioned in the Introduction.

In the final analysis, how strong or weak the domain theory needs to be can only be determined by reference to the needs of clinical practice, which is where the requirements come from. The authors of this paper are not qualified to pronounce definitively on this matter. Instead, we make some plausible assumptions, and examine how these affect the formal development. This requirements issue resurfaces at a number of points below.

6.5. Boundary-Centric Heartrate-Aware ASM Sensing

In Section 6.3, we directly specified the sensing in terms of the set of R values that could best describe the sensed electrogram-so-far egi in terms of $egmWin_R$ over continuous time. Of course, $egmWin_R$ is set valued, so it is not very amenable to an ultimate digital representation. In this section we recast the representation of $egmWin_R$ in terms of its upper and lower limits, a representation much more amenable to ultimate digital manipulation. Thus we get our next model, the BVH model. For the BVH model, we define the following minimum and maximum continuous variables:

$$winRmin_B(t) = \min winR(t) \tag{24}$$

$$winRmax_B(t) = \max winR(t) \tag{25}$$

The BVH model is framed in terms of the variables $winRmin_B$ and $winRmax_B$, and the dev variable is dispensed with. We define the constants:

$$R_{MAX,th} = R_{MAX} + \tfrac{1}{2}(T_{MAX} - R_{MAX}) \tag{26}$$

$$R_{0,th} = R_0 - \tfrac{1}{2}(R_0 - T_0) \tag{27}$$

observing that $T_0 < R_{0,th} < R_0$ and $R_{MAX} < R_{MAX,th} < T_{MAX}$ will hold. As the rules below make clear, in the BVH model, the values $R_{0,th}$ and $R_{MAX,th}$ are used as flags for ‘bad’ electrogram behaviour. In this and in subsequent models introduced below, for ease of subsequent comparison, we subscript corresponding variables to distinguish them from earlier predecessors. Mode and bound variables remain unaffected.

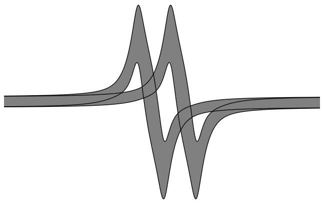

Fig. 6. Illustrating the impossibility of a disconnected $egmWin_R$ set during sensing.

Initialisation is as follows:

$$\begin{aligned}
&\textsc{Init}_{BVH} = \\
&\textbf{if}\ \ t = 0\ \ \textbf{then} \\
&\quad \textbf{do}\ \ mode\ := \ refractory, \\
&\qquad pulseGOOD\ := \ \mathsf{false},\ pulseBAD\ := \ \mathsf{false}, \\
&\qquad winRmin_B\ := \ R_{MAX},\ winRmax_B\ := \ R_0, \\
&\qquad egirise_B\ := \ 0,\ egifall_B\ := \ 0
\end{aligned} \tag{28}$$

The refractory period is unaffected:

$$\textsc{Refractory}_{BVH} \ \stackrel{\text{c}}{=}\ \ \textbf{if}\ \ mode = refractory\ \ \textbf{then do}\ \ \mathsf{skip} \tag{29}$$

The start of sensing is modified thus:

$$\begin{aligned}
&\textsc{StartSensing}_{BVH}\ = \\
&\textbf{if}\ \ mode = refractory \wedge t = T_0\ \ \textbf{then} \\
&\quad \textbf{do}\ \ mode\ := \ sensing, \\
&\qquad pulseGOOD\ := \ \mathsf{false},\ pulseBAD\ := \ \mathsf{false}, \\
&\qquad winRmin_B\ := \ R_{MAX},\ winRmax_B\ := \ R_0, \\
&\qquad egirise_B\ := \ 0,\ egifall_B\ := \ 0
\end{aligned} \tag{30}$$

The pliant rule of sensing is as follows:

$$\begin{aligned}
&\textsc{Egm}_{BVH}(\textbf{in}\ egi_B(t))\ \stackrel{\text{c}}{=} \\
&\textbf{let}\ \ wR(\bar{s}) = \boldsymbol{if}\ \bar{s} \notin \mathrm{dom}\ egi_B\ \boldsymbol{then}\ \emptyset \\
&\qquad\qquad \boldsymbol{else}\ \{R \mid R_0 \le R \le R_{MAX} \wedge \\
&\qquad\qquad\qquad\qquad \forall \tilde{s} \bullet t_{\mathrm{L}(t)} < \tilde{s} \le \bar{s} \Rightarrow egi(\tilde{s}) \in egmWin_R(\tilde{s})\}\ \boldsymbol{fi}\ \ \textbf{in} \\
&\textbf{if}\ \ mode = sensing\ \ \textbf{then} \\
&\quad \textbf{do}\ \ winRmin_B(t)\ := \ \min(\{R_{MAX,th}\} \cup wR(t)), \\
&\qquad winRmax_B(t)\ := \ \max(\{R_{0,th}\} \cup wR(t)), \\
&\qquad egirise_B(t)\ := \ \textbf{solve}\ \ \mathcal{D}\,egirise_B(t) = \mathcal{D}\,egi_B(t)\,\Theta(\mathcal{D}\,egi_B(t)), \\
&\qquad egifall_B(t)\ := \ \textbf{solve}\ \ \mathcal{D}\,egifall_B(t) = \mathcal{D}\,egi_B(t)\,\Theta(-\mathcal{D}\,egi_B(t))
\end{aligned} \tag{31}$$

$$\begin{aligned}
&\textsc{PulseGood}_{BVH} = \\
&\textbf{if}\ \ mode = sensing \wedge T_0 < t < T_{MAX} \wedge 0 < winRmax_B - winRmin_B < \delta_{th} \wedge \\
&\quad |egirise_B + egifall_B| < \delta_{rf} \wedge egirise_B > \Delta_{th}\ \ \textbf{then} \\
&\quad \textbf{do}\ \ mode\ := \ refractory, \\
&\qquad pulseGOOD\ := \ \mathsf{true}
\end{aligned} \tag{32}$$

$$\begin{aligned}
&\textsc{PulseBad}_{BVH} = \\
&\textbf{if}\ \ mode = sensing \wedge t = T_{MAX} \wedge [winRmax_B = R_{0,th} \vee winRmin_B = R_{MAX,th}]\ \ \textbf{then} \\
&\quad \textbf{do}\ \ mode\ := \ refractory, \\
&\qquad pulseBAD\ := \ \mathsf{true}
\end{aligned} \tag{33}$$

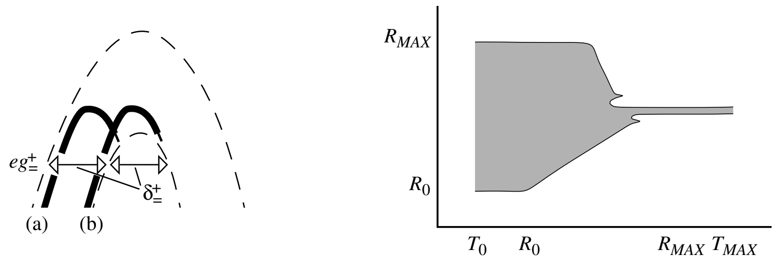

Fig. 7. Complex behaviour of $winR(t)$ when sensing a normal electrogram. On the left, two possible orientations of a normal electrogram-so-far and its extension with respect to the permitted window, illustrating a bifurcation in the set of permitted R values in *winR*. On the right, a schematic illustration of the possible subsequent behaviour of $winR(t)$, for t between T_0 and T_{MAX} for a normal electrogram..

6.6. On EGM$_{CVH}$ and EGM$_{BVH}$

In this section we consider the relationship between the EGM$_{CVH}$ and EGM$_{BVH}$ models. It turns out to be more complicated than it might initially appear.

Firstly, consider Fig. 6 which shows two dispositions in time of $egmWin_R$ (corresponding to two R values, R_1 and R_2 say) sufficiently separated in time that their union exhibits various holes, i.e. there are times t at which $egmWin_{R_1}(t)$ and $egmWin_{R_2}(t)$ do not intersect. We argue that for a normal electrogram, this situation cannot arise. The reason is that the electrogram is single-valued. Therefore, for an arbitrary time t, if the electrogram value is $egm(t)$, then for every R in $winR(t)$, it must hold that $egm(t) \in egmWin_R(t)$.

Noting the preceding, it is very tempting to think that for any t, the set $winR(t)$ is just all the values of R between the smallest element of $winR(t)$ and the largest element of $winR(t)$ — exactly that which is captured in the EGM$_{BVH}$ representation. However this is wrong. Fig. 7 shows what can happen.

On the left of Fig. 7 there are two possible orientations (with respect to the shape of the permitted window $egmWin_R$ — where $egmWin_R$ itself is shown using dashed curves in the figure) of an initial segment of a candidate electrogram-so-far, corresponding to an initial time t_1 say. These are the thick parts of curves (a) and (b), and they lead to a relatively large set of permitted R values in $winR(t_1)$, since it is clear that any position *in between* positions (a) and (b), of the electrogram-so-far (relative to the dashed curves), is also acceptable.

The electrogram-so-far is then extended a little, shown in the thin extensions to the thick curves (a) and (b), corresponding to a later time $t_2 > t_1$ say. The extensions to (a) and (b) sit either side of the bump in the lower boundary of the permitted window $egmWin_R$. It is now clear that *not every* position in between positions (a) and (b) of the electrogram-so-far is acceptable — there are intermediate positions that would cross the lower boundary of $egmWin_R$ before time t_2 was reached. Thus between t_1 and t_2, the set *winR* develops a hole, splitting into two disjoint pieces.

Of course, if the electrogram subsequently progresses according to a shape similar to *pulse*, then the orientation corresponding to curve (a) becomes infeasible as the descending electrogram inevitably crosses the lower boundary of $egmWin_R$. Thus, as time progresses, for a normal electrogram, the subset of R values corresponding to the orientation of curve (a), shrinks to a single point, and then vanishes, causing a sharp discontinuity in the size of the *winR* set, and leaving a much smaller *winR* centred on the curve (b) orientation for the remainder of the sensing. The early part of the right side of Fig. 7 illustrates this phenomenon schematically.

As a normal electrogram subsequently descends down the steep middle part of a *pulse*-like shape, it approaches the dip at the lowest point of the curve. In principle, an entirely analogous situation may take place there, except that with *winR* already curtailed from before, the dimensions of the hole and the subsequent discontinuity in the size of *winR*, are smaller. This is illustrated in the remainder of the right side of Fig. 7.

It has to be said that the situation described is not inevitable, but depends rather sensitively on the precise shape and width of the $egmWin_R$ window, and on the shape of the electrogram-so-far. Nevertheless, it cannot be excluded *a priori*.

Restricting our attention to just the first discontinuity, despite the preceding remarks, once an electrogram has progressed far enough along a *pulse*-like shape for the (a) configuration discussed above to have become infeasible, the situation cannot arise any more, i.e. the *winR* set will be connected, becoming a closed interval of R values, as it is most of the time. This requires the electrogram to have descended far enough down the steepest central part of *pulse*, that the difference in time between the value reached and the earlier moment when the electrogram reached the same value on its ascent, exceeds the width in time of $egmWin_R$ at that point.

In more detail, if an electrogram is to remain within $egmWin_R$ (for some R) throughout the heartbeat, we need to be sure it has traversed the 'bump' shown in the left hand side of Fig. 7, before we can be sure that *winR* will be connected. To this end, for every electrogram value *eg* that is reached, let t_a^{eg} be the last time that *eg* is visited as the electrogram advances up the left part of the bump, and let t_b^{eg} be the first time that *eg* is visited as the electrogram descends down the right part of the bump. Let $eg^+_=$ be the electrogram value such that the width in time of $egmWin_R$ at value $eg^+_=$, is a duration $\delta^+_=$, equal to the duration of the interval between the left and right parts of the bump. (The value $eg^+_=$, and the intervals of length $\delta^+_=$ are illustrated in the left part of Fig. 7.) Then we can be sure that *winR* will have become connected again by the time we have reached $t_b^{eg^+_=}$, which is the time of the first visit of the electrogram to value $eg^+_=$ as it descends down the right part of the bump — because it is evidently true that $t_b^{eg^+_=} - t_a^{eg^+_=} > \delta^+_=$, and so the part of the electrogram between $t_a^{eg^+_=}$ and $t_b^{eg^+_=}$ could not fit into the left hand part of the bump without exiting the permitted zone. (Analogous considerations, involving a value $eg^-_=$ and a duration $\delta^-_=$, concern the dip at the lowest point of *pulse*.)

These remarks all show that the interval between minimum and maximum values in *winR* at a given time t may be a proper superset of $winR(t)$. The shape of $egmWin_R$ indicates that this possibility disappears as the strongly varying part of $egmWin_R$ is passed. This scenario has practical implications for the relationship between the *CVH* and *BVH* models, since the minimum and maximum values $winR(t)$ alone are insufficient to define $winR(t)$ completely. This in turn implies that the formal relationship between the *CVH* and *BVH* models either needs to allow for the passage of sufficient time before making any claims that can be affected by such transient behaviour, or needs to restrict itself to claims that are insensitive to these effects.

In general such phenomena can prove challenging to formalisms that insist on establishing a finegrained relationship between models in a development hierarchy. However, in working with ASM, we are in a better position, since the granularity of the fragments which are related via ASM refinements is allowed to vary quite widely.

Accordingly, we are able to set up a refinement between the *CVH* and *BVH* models which averts its gaze from those parts of the two models' detailed behaviours that are hard to make precise, whether for reasons of complexity or of incomplete knowledge of the application domain (to the extent that would be needed to derive mathematically rigorous finegrained statements).

Central to the refinement will be a $(1,1)$ diagram covering the runtime execution of EGM_{CVH} and EGM_{BVH}, and central to such a refinement is the retrieve relation. We can write down a suitable candidate straight away:

$$\begin{aligned}
&R_{CVH,BVH}(\langle winR, dev, egirise, egifall\rangle, \\
&\qquad\quad \langle winRmin_B, winRmax_B, egirise_B, egifall_B\rangle) \;\equiv \\
&\quad egirise = egirise_B \wedge egifall = egifall_B \wedge \\
&\quad (dev \neq 0 \Rightarrow ((\exists s \bullet winR = \{s\}) \wedge [winRmin_B \ldots winRmax_B] = \emptyset)) \wedge \\
&\quad (dev = 0 \Rightarrow winR = [winRmin_B \ldots winRmax_B])
\end{aligned} \tag{34}$$

The retrieve relation (34) expresses the equalities of $egirise/egirise_B$ and $egifall/egifall_B$, which are obvious since they are defined in identical ways. Also a nonzero dev corresponds to $winR$ having been reduced to a single point according to (17), which in turn corresponds to $[winRmin_B \ldots winRmax_B]$ being empty according to (31). The final line of (34) expresses the equality of $winR$ and the interval $[winRmin_B \ldots winRmax_B]$ when $dev = 0$. By our earlier discussion, this last equality will not hold throughout a pair of $\textsc{Egm}_{CVH}$ and $\textsc{Egm}_{BVH}$ pliant transitions, but we can be sure of it at either end of the pair, which is all that we need, by the discussion surrounding (6).

It is sufficient to define the input relation of the refinement $In_{CVH,BVH}$, to be an equality between egi and egi_B, since they are equal by definition. And we do not need an output relation since there are no outputs.

6.7. Discretizing Continuous Models

Now we turn our attention to the fact that realistic sensing works in a discrete manner. Unlike a traditional discrete transition, which only manipulates a small amount of information, from a discrete information point of view, a pliant transition like (13) contains an infinite amount of information, i.e. the value of $egm(t)$ at each $t \in (T_0 \ldots T_{MAX})$ — all this despite the fact that $\textsc{Egm}_A$ is nevertheless finitely specified. To make use of this large amount of information, and to relate it to a framework (such as a model based development framework) which is focused on discrete pieces of information, we have to be specific about which parts of the underlying information we are going to use, and how.

The simplest approach to this issue comes via refinement. We can take a relatively long-lived pliant transition, such as $\textsc{Egm}_{BVH}$, and refine it via a $(1,n)$ diagram into a sequence of alternating mode and (short-lived) pliant transitions, designed such that their net effect is, in essence, equivalent to the original long-lived transition. We ensure that the short-lived pliant transitions are sufficiently short-lived that the changes in the values of their variables are sufficiently small that their effects can be catered for at a system level by focusing on just the mode transitions. Doing this properly, enables us thereafter to discard the pliant transitions, leaving us with a purely discrete transition system. This can be further refined towards implementation if necessary.

The strategy just described works best in either a *sensing* or a *control* scenario. In the former case, the relevant variables are read-only, in the latter case write-only. In such cases, the fact that there are influences on these variables other than those described in the purely discrete model causes no problem. This contrasts with the case of state variables, which are both read and written within the same collection of rules, and where simply discarding the pliant transitions could be more difficult to justify.

6.8. Discretized Heartrate-Aware ASM Sensing

In the *BVH* model, sensing was continuous. Following (10), adapted to the interval based monitoring in (31), this can be phrased as the question:

$$
\begin{aligned}
&(\exists R \bullet R_0 \le R \le R_{MAX} \wedge (\forall t \bullet T_0 < t < T_{MAX} \Rightarrow \\
&\qquad egi(t) \overset{?}{\in} [winRmin_B(t) \ldots winRmax_B(t)]))
\end{aligned}
\tag{35}
$$

Turning to discretized sensing, which will be captured in the *DVH* model, we assume that the sensing policy is such that the pacemaker observes the patient's electrogram at the times $t_n = T_0 + n\delta_t$, where $n \in 1 \ldots N$ and $T_0 + N\delta_t = T_{MAX}$. As before, if a normal heartbeat is detected, then this sequence is cut short.

In the *DVH* model, the continuous time question (35) is replaced by the sampled time question:

$$
\begin{aligned}
&(\exists R \bullet R_0 \le R \le R_{MAX} \wedge (\forall n \bullet 1 \le n \le N \Rightarrow \\
&\qquad egi(t_n) \overset{?}{\in} [winRmin_B(t_n) \ldots winRmax_B(t_n)]))
\end{aligned}
\tag{36}
$$

Note that although (35) and (36) have a comparable syntactic complexity, this conceals a much larger semantic difference. If we unravel the universal quantification in (36), we obtain an expression roughly N times larger. However, if we try to unravel the universal quantification in (35) we cannot do so, because it refers to a continuous infinity of values. So (35) contains much more information than (36), making (36) a weakening of (35). An affirmative answer to (35) implies an affirmative answer to (36), but the converse is not true. The issue then arises as to the relationship between the questions (35) and (36), and to their respective answers. We discuss this in more detail below.

Here then is the *DVH* model. Initialisation is largely as before, though the variables are subscripted with D and we now have $egms_D$, which holds the discretely sensed electrogram values and is initialised to the empty sequence:[4]

$$\begin{array}{l}
\textsc{Init}_{DVH} = \\
\textbf{if}\ \ t = 0\ \ \textbf{then} \\
\quad \textbf{do}\ \ mode\ := \ refractory, \\
\quad\quad pulseGOOD\ :=\ \mathsf{false},\ pulseBAD\ :=\ \mathsf{false}, \\
\quad\quad winRmin_D\ :=\ R_{MAX},\ winRmax_D\ :=\ R_0, \\
\quad\quad egirise_D\ :=\ 0,\ egifall_D\ :=\ 0, \\
\quad\quad egms_D\ :=\ \langle\rangle
\end{array} \qquad (37)$$

The refractory period is largely unchanged:

$$\textsc{Refractory}_{DVH} \stackrel{c}{=}\ \ \textbf{if}\ \ mode = refractory\ \ \textbf{then do}\ \ \mathsf{skip} \qquad (38)$$

The start of sensing is also similar:

$$\begin{array}{l}
\textsc{StartSensing}_{DVH}\ = \\
\textbf{if}\ \ mode = refractory \wedge t = T_0\ \ \textbf{then} \\
\quad \textbf{do}\ \ mode\ :=\ sensing, \\
\quad\quad pulseGOOD\ :=\ \mathsf{false},\ pulseBAD\ :=\ \mathsf{false}, \\
\quad\quad winRmin_D\ :=\ R_{MAX},\ winRmax_D\ :=\ R_0, \\
\quad\quad egirise_D\ :=\ 0,\ egifall_D\ :=\ 0, \\
\quad\quad egms_D\ :=\ \langle\rangle
\end{array} \qquad (39)$$

Sensing itself now consists of alternating pliant and mode transitions. The pliant transitions are basically trivial, in line with our intention to discard them later.

$$\textsc{Egm}_{DVH}(\textbf{in}\ egi_D(t)) \stackrel{c}{=}\ \ \textbf{if}\ \ mode = sensing\ \ \textbf{then do}\ \ \mathsf{skip} \qquad (40)$$

The hard work of discretized sensing comes in the mode transitions that interleave successive occurrences of $\textsc{Egm}_{DVH}$.

$$\begin{array}{l}
\textsc{EgmD}_{DVH}(\textbf{in}\ egi_D)\ = \\
\textbf{let}\ \ n = (t - T_0)/\delta_t\ \ \textbf{in} \\
\textbf{let}\ \ egms'_D = egms_D :: egi_D\ \ \textbf{in} \\
\textbf{let}\ \ wR_D(\bar{m}) = \textbf{\textit{if}}\ \bar{m} \notin \mathrm{dom}\, egms'_D\ \textbf{\textit{then}}\ \emptyset \\
\qquad\qquad\qquad \textbf{\textit{else}}\ \{R \mid R_0 \le R \le R_{MAX} \wedge \\
\qquad\qquad\qquad\qquad \forall \tilde{m} \bullet 1 \le \tilde{m} \le \bar{m} \Rightarrow egms'_D(\tilde{m}) \in egmWin_R(t_{\tilde{m}})\}\ \textbf{\textit{fi}}\ \ \textbf{in} \\
\textbf{if}\ \ mode = sensing \wedge n \in \mathbb{N} \wedge 1 \le n \le N\ \ \textbf{then} \\
\quad \textbf{do}\ \ winRmin_D\ :=\ \min(\{R_{MAX,th}\} \cup wR(n)), \\
\quad\quad winRmax_D\ :=\ \max(\{R_{0,th}\} \cup wR(n)), \\
\quad\quad egirise_D\ :=\ egirise_D + \Phi(egi_D > \mathrm{last}\, egms_D)\,(egi_D - \mathrm{last}\, egms_D), \\
\quad\quad egifall_D\ :=\ egifall_D + \Phi(egi_D < \mathrm{last}\, egms_D)\,(egi_D - \mathrm{last}\, egms_D), \\
\quad\quad egms_D\ :=\ egms'_D
\end{array} \qquad (41)$$

In (41), the sensed electrogram values at the times t_n are assembled into the sequence $egms_D$. These values are used in $\textsc{EgmD}_{DVH}$ for a discretized version of the sensing calculation in $\textsc{Egm}_{BVH}$. Thus, instead of comparing the sensed electrogram values at all times within the sensed period-so-far, we

[4]For convenience, in this paper sequences are indexed starting from 1.

only inspect them at the discrete moments t_n falling within that period. As before, the sensed electrogram values determine the most permissive range of R values such that the sensed electrogram-so-far is within $[winRmin_D(t_n)\ldots winRmax_D(t_n)]$ for all relevant t_n. In (41), the function Φ returns 1 when its boolean argument is true, else it returns 0.

Note the contrast between $\textsc{Egm}_{BVH}$ and $\textsc{EgmD}_{DVH}$. In $\textsc{EgmD}_{DVH}$ we can store the accumulated *egi* values into a sequence, because a sequence is a discrete data structure. In $\textsc{Egm}_{BVH}$ the *egi* values form an arbitrary continuum, and although there is nothing in the type system of our extended ASM to prevent us from defining variables whose values are the *whole graph* of an arbitrary continuous function, we consider it unreasonable to do so, except in a very abstract non-implementable model.

In fact in $\textsc{Egm}_{BVH}$ we have the whole of the input *egi* available, so we do not need an extra state variable for this information. By contrast, in $\textsc{EgmD}_{DVH}$, not only do we not have the whole of the input available, but we also regard $\textsc{EgmD}_{DVH}$ as being 'rather closer to implementable', so we take care that its variables are 'close to implementable' too. Of course there is no problem in defining variables that hold *individual instantaneous values* of continuous functions, at any level of abstraction, and, up to a point, we compensate for the self-imposed lack of availability of the graph of a continuous function by allowing instantaneous values of its *derivative* (where the derivative can be assumed to exist), as we already did in (17). To a great extent, reasoning about the derivative of a continuous function in the continuous case, plays a role analogous to that of reasoning about consecutive values of a sequence in the discrete case.

Note that in $\textsc{PulseGood}_{DVH}$ and $\textsc{PulseBad}_{DVH}$ we use '$(\exists n \bullet t = t_n)$' as a shorthand for the more explicit stipulation of time as an integral multiple of δ_t calculated in $\textsc{EgmD}_{DVH}$. We use the same shorthand in the remainder of the paper when convenient.

$$
\begin{array}{l}
\textsc{PulseGood}_{DVH} = \\
\textbf{if}\ \ mode = sensing \wedge dev_D = 0 \wedge (\exists n \bullet t = t_n) \wedge T_0 < t < T_{MAX}\ \wedge \\
\quad 0 < winRmax_D - winRmin_D < \delta_{th} \wedge |egirise_D + egifall_D| < \delta_{rf} \wedge \\
\quad egirise_D > \Delta_{th}\ \ \textbf{then} \\
\quad \textbf{do}\ \ mode\ := \ refractory, \\
\qquad pulseGOOD\ :=\ \mathsf{true}
\end{array}
\tag{42}
$$

$$
\begin{array}{l}
\textsc{PulseBad}_{DVH} = \\
\textbf{if}\ \ mode = sensing \wedge dev_D > 0 \wedge t = T_{MAX} \wedge \\
\quad [winRmax_D = R_{0,th} \vee winRmin_D = R_{MAX,th}]\ \ \textbf{then} \\
\quad \textbf{do}\ \ mode\ := \ refractory, \\
\qquad pulseBAD\ :=\ \mathsf{true}
\end{array}
\tag{43}
$$

Observe that according to ASM semantics, sensing is disabled by the simultaneous execution of $\textsc{EgmD}_{DVH}$ together with one of $\textsc{PulseGood}_{DVH}$ or $\textsc{PulseBad}_{DVH}$.

6.9. On $\textsc{Egm}_{BVH}$, the B^*VH model, and Finegrained Observation

The move from the *BVH* to the *DVH* model exemplifies the discretization step needed to transform a model of some aspects of the real physical world into a model that can connect with a digitally implemented system. Making rigorous statements about the relationship between the two models helps to combine them into a single integrated and consistent development path, and also underpins the quality of the modeling. Showing how to make such a connection is the concern of this and the next section.

Our starting point is to modify the *BVH* model, yielding the B^*VH model, by adding the following innocuous mode transition:

$$
\begin{array}{l}
\textsc{EgmD}_{BVH}(\textbf{in}\ egi) = \\
\textbf{if}\ \ mode = sensing \wedge (\exists n \bullet t = t_n \wedge n \in \mathbb{N} \wedge 1 \leq n < N) \wedge \\
\quad \neg(0 < winRmax_D - winRmin_D < \delta_{th} \wedge |egirise_D + egifall_D| < \delta_{rf} \wedge \\
\qquad egirise_D > \Delta_{th})\ \ \textbf{then} \\
\quad \textbf{do}\ \ \mathsf{skip}
\end{array}
\tag{44}
$$

This, the *BVH* model analogue of EGMD_{DVH}, is enabled at all integral multiples of δ_t prior to the enabling of PULSEGOOD_{BVH} or PULSEBAD_{BVH}. Since it skips, it merely interrupts EGM_{BVH} momentarily, which immediately resumes from where it left off, and since its guards have been designed not to overlap with those of PULSEGOOD_{BVH} and PULSEBAD_{BVH}, it does not interfere with their working either. Thus the behaviours of the *BVH* and *B*VH* models are identical in essence.

The purpose of introducing the *B*VH* model is to ease the relationship between the *BVH* and *DVH* models. We first relate the *BVH* model to the *B*VH* model, and then go from *B*VH* to *DVH*.

The relationship between the *BVH* model and the *B*VH* model is straightforward. Aside from rules which are identical, there is a $(1,2m-1)$ ASM refinement diagram from a single execution of EGM_{BVH} in *BVH* to a sequence of m executions of EGM_{BVH}, interleaved by $m-1$ executions of EGMD_{BVH} in *B*VH*. Here, m ranges from the minimum number needed to reach the earliest possible enabling of PULSEGOOD_{BVH} (for rapid normal heartbeats), to N (for abnormal cardiac activity). The retrieve, input and output relations for this refinement are just the obvious identities.

The creation of the *B*VH* model introduces a notion of finegrained observation to our systems. At fairly high levels of abstraction, where long-lived pliant transitions are commmon, the formulation of ASM refinement and retrenchment that we gave, only enables the *whole* of such a long-lived pliant transition to be related to the steps of a lower-level more finegrained model. However, if we break up a long-lived pliant transition by the introduction of trivial mode transitions as above, we subsequently have the option of relating the resulting short-lived steps much more closely to a lower-level finegrained model.

6.10. On the *B*VH* and *DVH* models

The introduction of the EGMD_{BVH} transitions in the *B*VH* model allows us to express the relationship between the core sensing elements of the *B*VH* and *DVH* models using a number of $(1,1)$ diagrams instead of one $(1,2m-1)$ diagram, exposing the technical details to more thorough scrutiny.

There are two non-trivial $(1,1)$ diagrams to consider: one from EGM_{BVH} to EGM_{DVH}, and another from EGMD_{BVH} to EGMD_{DVH}. We start with the first of these.

The relationship between EGM_{BVH} and EGM_{DVH} (and thus the relationship between the *B*VH* and the *DVH* models as a whole) will be phrased as a retrenchment — since EGM_{BVH} follows its input quite closely while EGM_{DVH} ignores its input, expecting a refinement would be unrealistic.

The retrenchment data starts with the retrieve relation. As before, we suppress the mode variables.

$$\begin{aligned}
&R_{BVH,DVH}(\langle winRmin_B, winRmax_B, egirise_B, egifall_B\rangle,\\
&\qquad\qquad \langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D\rangle) \;\equiv\\
&\quad [winRmin_B \dots winRmax_B] \subseteq [winRmin_D \dots winRmax_D] \;\wedge\\
&\quad (egirise_D > 0 \Rightarrow egirise_B > 0) \wedge (egifall_D < 0 \Rightarrow egifall_B < 0)
\end{aligned} \tag{45}$$

Noting that the *DVH* interval of R values is specified by a subset of the constraints that specify the *BVH* interval, the stated inclusion will hold. Also, since EGM_{BVH} monitors *egi* continuously whereas EGM_{DVH} only monitors *egi* sporadically, analogous remarks apply to $egirise_D$ versus $egirise_B$ and $egifall_D$ versus $egifall_B$. So we expect all the clauses of $R_{BVH,DVH}$ to hold unconditionally.

Next we have the within relation, $W_{\text{EGM}_{BVH},\text{EGM}_{DVH}}$, in (46). The first clause of $W_{\text{EGM}_{BVH},\text{EGM}_{DVH}}$, simply states the equality of the inputs to the continuous and discretized models.

The objective of the second clause is to formalise our assumption that the sampling frequency is high enough. Thus, if we know the values of the input signal at the beginning and end of a sampling period of length δ_t, then nothing untoward will flow from our ignorance of the input signal in its interior. In this instance, 'nothing untoward' concerns the quality of the estimates of the R values that can be used to describe the observed electrogram over the sampling period.

Now the set of suitable R values can only decrease with time, since as time progresses, the accumulating values of the observed electrogram can only add new constraints on the acceptable R values without removing any earlier ones. Therefore, if the electrogram from $t_{\mathrm{L}(t)}$ up to $t_{\mathrm{R}(t)}$ is capable of being described by $egmWin_R$ for some suitable R, then any such R is bound to be in the *DVH* interval of R values at $t_{\mathrm{L}(t)}$.

$$
\begin{aligned}
& W_{\mathrm{EGM}_{BVH},\mathrm{EGM}_{DVH}}(egi_B(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)})), egi_D(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)})), \\
& \qquad \langle winRmin_B, winRmax_B, egirise_B, egifall_B \rangle (t_{\mathrm{L}(t)}), \\
& \qquad \langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D \rangle (t_{\mathrm{L}(t)})) \ \equiv \\
& \quad (\forall t \bullet t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)}) \Rightarrow egi_B(t) = egi_D(t)) \ \wedge \\
& \quad [\ (\exists R \bullet R \in [winRmin_D(t_{\mathrm{L}(t)}) \ldots winRmax_D(t_{\mathrm{L}(t)})] \wedge egi_D(t_{\mathrm{R}(t)}) \in egmWin_R(t_{\mathrm{R}(t)})) \\
& \qquad \Rightarrow \\
& \quad\ (\exists R \bullet R \in [winRmin_B(t_{\mathrm{R}(t)}) \ldots winRmax_B(t_{\mathrm{R}(t)})] \ \wedge \\
& \qquad (\forall t \bullet t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)}) \Rightarrow egi(t) \in egmWin_R(t)))\ ]
\end{aligned} \tag{46}
$$

What our assumption about the behaviour of the electrogram amounts to, is the inference that **if** the electrogram is well behaved at both the beginning and end of the sampling period, **then** it is well behaved throughout the sampling period, *also as far as the continuous model is concerned*. This is the content of the second clause of $W_{\mathrm{EGM}_{BVH},\mathrm{EGM}_{DVH}}$. Unlike the facts stated in the retrieve relation, which we argued hold unconditionally, it has to be an *assumption* since the electrogram comes from the environment and is not controlled by the sensing system. (Moreover, in the light of earlier discussions, it would not be hard to invent a counterexample that violated this assumption.)

Since it is an assumption, we always take the second clause to be true. However, since it is an implication, it can be true in more than one way. On the one hand, if its own assumptions are true, the electrogram is well behaved throughout the sampling period. In this case, we are continuing on course for a normal heartbeat, in which the *DVH* interval of acceptable R values continues to be nonempty. This is expressed in the output relation:

$$
\begin{aligned}
& O_{\mathrm{EGM}_{BVH},\mathrm{EGM}_{DVH}}([\langle winRmin_B, winRmax_B, egirise_B, egifall_B \rangle](t_{\mathrm{L}(t)}, t_{\mathrm{R}(t)}), \\
& \qquad egi(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)})), \\
& \qquad [\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D \rangle](t_{\mathrm{L}(t)}, t_{\mathrm{R}(t)}), \\
& \qquad egi_D(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)}))) \ \equiv \\
& \quad winRmin_B < winRmax_B
\end{aligned} \tag{47}
$$

On the other hand, the assumptions of the implication may be false. In that case, a fact witnessed by the inversion of the values of $winRmin_B$ and $winRmax_B$, which is the content of the concedes relation:

$$
\begin{aligned}
& C_{\mathrm{EGM}_{BVH},\mathrm{EGM}_{DVH}}([\langle winRmin_B, winRmax_B, egirise_B, egifall_B \rangle](t_{\mathrm{L}(t)}, t_{\mathrm{R}(t)}), \\
& \qquad egi(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)})), \\
& \qquad [\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D \rangle](t_{\mathrm{L}(t)}, t_{\mathrm{R}(t)}), \\
& \qquad egi_D(t \in (t_{\mathrm{L}(t)} \ldots t_{\mathrm{R}(t)}))) \ \equiv \\
& \quad winRmin_B > winRmax_B
\end{aligned} \tag{48}
$$

The absence of the equality case in either of (47) or (48) correlates with our assumption that normal and abnormal heartbeats can be cleanly distinguished. This completes the discussion of the $(1,1)$ diagram from EGM_{BVH} to EGM_{DVH}.

Now we consider the $(1,1)$ diagram from EGMD_{BVH} to EGMD_{DVH}. The retrieve relation continues to be (45) of course, since a single retrieve relation has to do duty for all the (m,n) diagrams in a refinement or retrenchment.

The within relation is:

$$W_{\text{EGMD}_{BVH},\text{EGMD}_{DVH}}(egi_B, egi_D,$$
$$\langle winRmin_B, winRmax_B, egirise_B, egifall_B\rangle,$$
$$\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D\rangle) \equiv$$
$$egi_B = egi_D \tag{49}$$

In the output and concedes relations which follow, we see a nice symmetry, in that $O/C_{\text{EGM}_{BVH},\text{EGM}_{DVH}}$ express what happens to the *BVH* model *R* value interval (since for EGM_{BVH} and EGM_{DVH} it is the continuous variable that is doing active work), while $O/C_{\text{EGMD}_{BVH},\text{EGMD}_{DVH}}$ express what happens to the *DVH* model interval (since for EGMD_{BVH} and EGMD_{DVH} it is the discretized variable that is doing active work).

$$O_{\text{EGMD}_{BVH},\text{EGMD}_{DVH}}(\langle winRmin_B, winRmax_B, egirise_B, egifall_B\rangle,$$
$$\langle winRmin'_B, winRmax'_B, egirise'_B, egifall'_B\rangle, egi,$$
$$\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D\rangle,$$
$$\langle egms'_D, winRmin'_D, winRmax'_D, egirise'_D, egifall'_D\rangle, egi_D) \equiv$$
$$winRmin'_D < winRmax'_D \tag{50}$$

$$C_{\text{EGMD}_{BVH},\text{EGMD}_{DVH}}(\langle winRmin_B, winRmax_B, egirise_B, egifall_B\rangle,$$
$$\langle winRmin'_B, winRmax'_B, egirise'_B, egifall'_B\rangle, egi,$$
$$\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D\rangle,$$
$$\langle egms'_D, winRmin'_D, winRmax'_D, egirise'_D, egifall'_D\rangle, egi_D) \equiv$$
$$winRmin'_D > winRmax'_D \tag{51}$$

Finally, we complete the relationship between the *B*VH* and *DVH* models by observing that aside from what we have done, the rules for INIT, REFRACTORY, STARTSENSING, PULSEGOOD, PULSEBAD, are identical in the two models. Therefore these transitions are related in the two models by a limiting form of retrenchment that amounts to a refinement, and which arises by defaulting the within, output and concedes relations of the retrenchment data for these rules to true, true, and false respectively.

6.11. An Algorithmic Sensing Model

Our final development step creates the *AVH* model, an algorithmic model that gets closer to calculating the minimum and maximum needed by the EGMD_{DVH} rule in the *DVH* model. We do this development purely at the discrete level, ignoring the pliant rules, paralleling the fact that beyond a certain point, once all the activity has been captured in mode transitions, further system development will naturally restrict itself to the world of purely discrete transition systems.

The main rule that is significantly affected at this level is the EGMD rule. However, its *AVH* version requires two new state variables, *minIdx* and *maxIdx*, which affects a couple of the other rules in small ways too.

Although written in a syntax resembling the earlier ASM notation, the rules of the *AVH* model are intended to be understood as conventional sequential programs, with sequential composition written as **;**, and with the syntax interpreted according to a relatively conventional ASM sequential program operational semantics. Connected with this is the understanding that the *AVH* rules are intended to be scheduled at suitable moments by a conventional external scheduler. To this end, some of the guards of the earlier models (particularly those concerning scheduling times) are converted into assertions in the *AVH* model, reflecting the fact that that it is no longer the responsibility of the rules themselves to only fire when their guards are true. This change affects the remaining mode rules of the system, so in the end, all of the mode rules are affected one way or another.

Here are the rules that are affected by the initialisation of the *minIdx* and *maxIdx* variables. In (52) and (53), N_{MAX} is a constant that we will discuss later.

$$\begin{array}{l} \text{I}\scriptstyle{\text{NIT}}_{AVH}\ = \\ \textbf{assert}\ t = 0\ ; \\ mode\ :=\ refractory\ ; \\ pulseGOOD\ :=\ \mathsf{false}\ ;\ \ pulseBAD\ :=\ \mathsf{false}\ ; \\ winRmin_A\ :=\ R_{MAX}\ ;\ \ winRmax_A\ :=\ R_0\ ; \\ egirise_A\ :=\ 0\ ;\ \ egifall_A\ :=\ 0\ ; \\ egms_A\ :=\ \langle\rangle\ ; \\ minIdx\ :=\ 1\ ;\ \ maxIdx\ :=\ N_{MAX} \end{array} \tag{52}$$

$$\begin{array}{l} \text{S}\scriptstyle{\text{TART}}\textstyle\text{S}\scriptstyle{\text{ENSING}}_{AVH}\ = \\ \textbf{assert}\ mode = refractory \wedge t = T_0\ ; \\ mode\ :=\ sensing\ ; \\ pulseGOOD\ :=\ \mathsf{false}\ ;\ \ pulseBAD\ :=\ \mathsf{false}\ ; \\ winRmin_A\ :=\ R_{MAX}\ ;\ \ winRmax_A\ :=\ R_0\ ; \\ egirise_A\ :=\ 0\ ;\ \ egifall_A\ :=\ 0\ ; \\ egms_A\ :=\ \langle\rangle\ ; \\ minIdx\ :=\ 1\ ;\ \ maxIdx\ :=\ N_{MAX} \end{array} \tag{53}$$

We note that in the *DVH* model, in the INIT_{DVH} and $\text{STARTSENSING}_{DVH}$ rules, the guards are simple functions of time and of the mode variables. We can assume that these are readily available to the scheduler, legitimising our conversion of their guards into assertions. However, when we turn our attention to the rules PULSEGOOD_{DVH} and PULSEBAD_{DVH}, we observe that their guards are not so simple, depending on more complex expressions of the state variables. We take the view that it is unreasonable to expect the scheduler to do the necessary evaluations, so we must convert them into something that can be invoked in a simple way by the scheduler, with the more complicated calculations done at runtime within conditional statements. Since both the PULSEGOOD and PULSEBAD rules are intended to run at some statically unknown multiple of t_δ (a multiple which is therefore unknowable *a priori* by the scheduler), and EGMD also runs at those moments, in the *AVH* model, we convert both rules into a routine, $\text{PULSEMAYBEGOODORBAD}_{AVH}$, which is called by EGMD_{AVH} at the end of its work, and which implements PULSEGOOD, or PULSEBAD, or does nothing, as appropriate.

$$\begin{array}{l} \text{PULSEMAYBEGOODORBAD}_{AVH}\ = \\ \textbf{if}\ T_0 < t < T_{MAX} \wedge 0 < winRmax_A - winRmin_A < \delta_{th} \wedge \\ \quad |egirise_A + egifall_A| < \delta_{rf} \wedge egirise_A > \Delta_{th} \\ \textbf{then}\ mode\ :=\ refractory\ ; \\ \qquad pulseGOOD\ :=\ \mathsf{true} \\ \textbf{elseif}\ t = T_{MAX} \wedge [winRmax_A = R_{0,th} \vee winRmin_A = R_{MAX,th}] \\ \textbf{then}\ mode\ :=\ refractory\ ; \\ \qquad pulseBAD\ :=\ \mathsf{true} \\ \textbf{else}\ \mathsf{skip} \\ \textbf{fi} \end{array} \tag{54}$$

Turning to the EGMD_{DVH} rule, we see that it contains assignments to the minimum variable $winRmin_D$ and to the maximum variable $winRmax_D$. We start by focusing on the minimum calculation. Our aim in the *AVH* model is to efficiently calculate the minimum on each call of EGMD_{AVH}. Since each call occurs after the introduction of exactly one additional electrogram observation, we can take advantage of the fact that the minimum acceptable R value for all the observations *prior to the most recent one* has been calculated during the previous call, and is available in the before-value of the variable holding the minimum ($winRmin_D$ in the *DVH* model).

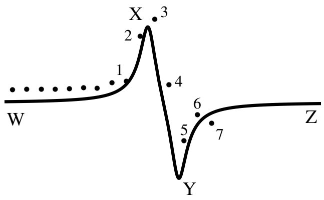

Fig. 8. Calculating the moving minimum as electrogram observations (represented as dots) become available. Points WXYZ partition the curve $egmWin^{-}_{R}$ into first rising (WX), falling (XY), and second rising (YZ) segments. Observation 1 is sufficient for determining the current minimum R value R_{pre} for $egmWin^{-}_{R}$, until observation 7 is added. Then R has to increase to accommodate it, moving the $egmWin^{-}_{R}$ curve to the right. Only electrogram observations above the XY portion need to be checked to ensure that they remain above the $egmWin^{-}_{R}$ curve.

Using the notations of EGMD_{DVH} in (31), we define $Rmin_{\text{pre}}$ and $Rmin_{\text{post}}$ via:

$$\begin{aligned} Rmin_{\text{pre}} \;\equiv\;& \min(\{R_{MAX,th}\} \cup \{R \mid R_0 \le R \le R_{MAX} \;\wedge \\ &\qquad \forall n \bullet n \in \operatorname{dom} egms_D \Rightarrow egms_D(n) \ge egmWin^{-}_{R}(t_n)\}) \qquad (55) \\ Rmin_{\text{post}} \;\equiv\;& \min(\{R_{MAX,th}\} \cup \{R \mid R_0 \le R \le R_{MAX} \;\wedge \\ &\qquad \forall n \bullet n \in \operatorname{dom} egms'_D \Rightarrow egms'_D(n) \ge egmWin^{-}_{R}(t_n)\}) \qquad (56) \end{aligned}$$

Apart from initialisation time, these are just names for the before- and after- values of $winRmin_D$ during a call of EGMD_{DVH}. (N. B. In (55), if $egms_D = \langle\rangle$ (which will be true on the first call of (41)), then $Rmin_{\text{pre}}$ evaluates to R_0.)

Since $egms'_D = egms_D :: egi_D$, the formula for $Rmin_{\text{post}}$ has to take into account precisely one additional constraint compared with the formula for $Rmin_{\text{pre}}$, so that:

$$Rmin_{\text{pre}} \le Rmin_{\text{post}} \qquad (57)$$

This means that $egmWin^{-}_{Rmin_{\text{post}}}$ is essentially a copy of $egmWin^{-}_{Rmin_{\text{pre}}}$ that has been shifted to the right by a potentially non-zero non-negative amount to accommodate the new electrogram observation.

Referring to Fig. 8, the time points WXYZ divide up the $egmWin^{-}_{R}$ curve into first rising (WX), falling (XY), and second rising (YZ) segments. As $egmWin^{-}_{R}$ moves to the right, any electrogram observation lying above a rising segment will remain above it since, by the definition of a rising segment being shifted to the right, for a given t, we have $egmWin^{-}_{R_{\text{post}}}(t) \le egmWin^{-}_{R_{\text{pre}}}(t)$.

Conversely, any electrogram observation lying above the falling segment need not necessarily remain above it as $egmWin^{-}_{R}$ moves to the right, since, by the definition of a falling segment being shifted to the right, for a given t, we have $egmWin^{-}_{R_{\text{post}}}(t) \ge egmWin^{-}_{R_{\text{pre}}}(t)$. Therefore, such electrogram observations need to be checked explicitly.

Also, near X and Y, electrogram observations may move between rising and falling segments as the $egmWin^{-}_{R}$ curve moves, so, if X_{pre} is the before-value of X, and Y_{post} is the after-value of Y, all electrogram observations occurring between X_{pre} and Y_{post} need to be checked, to confirm whether they still lie above $egmWin^{-}_{R_{\text{post}}}$ after observation egi_B has been taken into account. If any do not, then the attempt to fit the observations to the desired shape fails, and ultimately, the pacemaker will have to generate an artificial depolarization.

Fig. 8 illustrates the process. Observations are accumulated, and, up to (and including) observation 6, all lie above the $egmWin^{-}_{R}$ curve determined by the value R_{pre}, which is the minimum R value forced by the requirement that observation 1 does not lie below $egmWin^{-}_{R}$ (this being the most stringent among all the requirements on R generated by all the observations up to 6 inclusive). Then observation 7 arrives, and it falls below $egmWin^{-}_{R_{\text{pre}}}$. So R has to be increased to a value R_{post} large enough so that observation 7 does not lie below $egmWin^-_{R_{\rm post}}$. In thus increasing R, there is a danger that the observations lying above the (pre- or post-) falling segment (let us say these are observations 3, 4, 5) may find themselves below the new position of the $egmWin^-_R$ curve — these therefore have to be checked. This reasoning leads to the code for $\textsc{EgmD}_{AVH}$ in (58).

$$
\begin{array}{l}
\textsc{EgmD}_{AVH}(\textbf{in}\ egi_A)\ =\\
\textbf{let}\ \ n = (t - T_0)/\delta_t\ \ ;\\
\textbf{assert}\ \ mode = sensing \wedge n \in \mathbb{N} \wedge 1 \le n \le N\ \ ;\\
egirise_A\ \ := \ egirise_A + \Phi(egi_A > \mathrm{last}\, egms_A)\,(egi_A - \mathrm{last}\, egms_A)\ \ ;\\
egifall_A\ \ := \ egifall_A + \Phi(egi_A < \mathrm{last}\, egms_A)\,(egi_A - \mathrm{last}\, egms_A)\ \ ;\\
egms_A := egms_A :: egi_A\ \ ;\\
\textbf{if}\ \ winRmin_A = R_{MAX,th}\ \ \textbf{then}\ \ \mathsf{skip}\\
\textbf{elseif}\ \ egi_A < egmWin^-_{MinValue}\ \ \textbf{then}\ \ winRmin_A\ \ := \ R_{MAX,th}\\
\textbf{elseif}\ \ egi_A \ge egmWin^-_{winRmin_A}(n\delta_t)\ \ \textbf{then}\ \ \mathsf{skip}\\
\textbf{else let}\ \ newRmin = n\delta_t - \textsc{SolveMin}(egi_A)\ \ ;\\
\qquad \textbf{if}\ \ \textsc{CheckMin}(newRmin, \mathrm{X}_{\rm pre}(winRmin_A), \mathrm{Y}_{\rm post}(newRmin))\\
\qquad \textbf{then}\ \ winRmin_A\ \ := \ newRmin\\
\qquad \textbf{else}\ \ winRmin_A\ \ := \ R_{MAX,th}\\
\qquad \textbf{fi}\\
\textbf{fi}\ \ ;\\
\textbf{if}\ \ winRmax_A = R_{0,th}\ \ \textbf{then}\ \ \mathsf{skip}\\
\textbf{elseif}\ \ egi_A > egmWin^+_{MaxValue}\ \ \textbf{then}\ \ winRmax_A\ \ := \ R_{0,th}\\
\textbf{elseif}\ \ egi_A \le egmWin^+_{winRmax_A}(n\delta_t)\ \ \textbf{then}\ \ \mathsf{skip}\\
\textbf{else let}\ \ newRmax = n\delta_t - \textsc{SolveMax}(egi_A)\ \ ;\\
\qquad \textbf{if}\ \ \textsc{CheckMax}(newRmax, \mathrm{X}_{\rm post}(newRmax), \mathrm{Y}_{\rm pre}(winRmax_A))\\
\qquad \textbf{then}\ \ winRmax_A\ \ := \ newRmax\\
\qquad \textbf{else}\ \ winRmax_A\ \ := \ R_{0,th}\\
\qquad \textbf{fi}\\
\textbf{fi}\ \ ;\\
\textsc{PulseMaybeGoodOrBad}_{AVH}
\end{array}
\tag{58}
$$

In the $\textsc{EgmD}_{AVH}$ code, after the assertion (which enforces the timing of calls to $\textsc{EgmD}_{AVH}$ as discussed above), any increment to the AVH variables $egirise_A$ and $egifall_A$ is recorded, before egi_A is appended to the existing electrogram observations. Then comes a case analysis for updating $egmWin^-_R$ (if necessary). Thus, if some earlier attempt to find a lower bound for all observations using $egmWin^-_R$ for an acceptable value of R has already failed ($winRmin_A = R_{MAX,th}$), then there is nothing more to be done.[5] Else, if the new observation egi_A is less than the lowest value, $egmWin^-_{MinValue}$, ever reached by $egmWin^-_R$ (which is independent of R) then failure occurs right now, and is signalled by setting $winRmin_A$ to $R_{MAX,th}$. Else, if the new observation egi_A is greater than or equal to the value of $egmWin^-_R$ for the current value of time and of R (this is $egmWin^-_{winRmin_A}(n\delta_t)$), then nothing need be done, since the current value of R for $egmWin^-_R$, i.e. $winRmin_A$, is adequate. Finally, if control falls through to the last case, then egi_A lies between $egmWin^-_{MinValue}$ and the current value of $egmWin^-_{winRmin_A}$. In that case, by the intermediate value theorem, since $egmWin^-_R$ (for any R) is continuous between these two values, then (again for any R), there is a solution t_R to the equation $egmWin^-_R(t_R) = egi_A$. This value t_R can be translated to a new R value, $newRmin$, that satisfies the equation $egmWin^-_{newRmin}(t) = egi_A$.

Routine $\textsc{SolveMin}$ (described below) finds the solution $\bar{t}$ to the equation $egmWin^-_0(\bar{t}) = egi_A$ (for the fixed R value 0). This is then translated into the new R value $newRmin$. Since the curve

[5]This point clarifies why we used values R_0 and R_{MAX} as intial values at the beginning of sensing (i.e. 'not yet gone bad'), and values $R_{MAX,th}$ and $R_{MAX,th}$ (i.e. 'definitely gone bad') to signal an abnormal heartbeat signal. Had we used the same values for both purposes, every call of $\textsc{EgmD}_{AVH}$ would skip here.

$egmWin_R^-$ must be shifted to the right, only *newRmin* values that are greater than $winRmin_A$ are useful, so SOLVEMIN begins its work by looking for a solution in the direction most likely to yield such a value. Routine CHECKMIN (described below) verifies this property of *newRmin*, and then checks all $egms_A$ values occurring between X_{pre} (which depends on $winRmin_A$, hence is written as $X_{pre}(winRmin_A)$) and Y_{post} (which similarly depends on *newRmin*, hence $Y_{post}(newRmin)$) to ensure that they lie above $egmWin^-_{newRmin}$.

Since, in general, the values X and Y are statically related to R, we do not describe further the calculation of $X_{pre}(winRmin_A)$ and $Y_{post}(newRmin)$, assuming these values known. Routine CHECKMIN returns true if all the checks pass, otherwise returning false. If CHECKMIN returns true, then $winRmin_A$ can be safely updated to *newRmin*, otherwise $winRmin_A$ is set to the failing value $R_{MAX,th}$. This completes the processing for the running minimum R value.

The reasoning for computing the running maximum is entirely analogous to the above, so we do not describe it in detail. Finally, EGMD_{AVH} calls $\text{PULSEMAYBEGOODORBAD}_{AVH}$ in case sensing can be terminated after this call.

We turn to the working of the auxiliary routines SOLVEMIN and CHECKMIN used in the minimum calculation. Since the job of SOLVEMIN is to invert the function $egmWin_R^-$, and $egmWin_R^-$ is not analytically invertible, SOLVEMIN must resort to a search-based strategy. We assume that SOLVEMIN has available a table, called $egmWinTab^-$ (most conveniently organised as an array indexed from 1 to N_{MAX}), of $(t, egmWin_0^-(t))$ pairs (the first and second components of which are addressed using the notations $egmWinTab^-.time$ and $egmWinTab^-.val$ respectively), with the t values selected so that linear interpolation can be used unproblematically. The state variable *minIdx* records the index of the last $egmWinTab^-$ entry used, on the assumption that the one needed for the next call of SOLVEMIN will usually be nearby.

Assuming that: (i) the first entry of $egmWinTab^-$ refers to the earliest value of $egmWin_0^-$ that could possibly be needed (i.e. the value at time $T_0 - R_{MAX}$); (ii) the last entry of $egmWinTab^-$ refers to the latest value of $egmWin_0^-$ that could possibly be needed (i.e. the value at time $T_{MAX} - R_0$); (iii) some $egmWinTab^-$ entry refers to the maximum $egmWin_0^-$ value at X; some $egmWinTab^-$ entry refers to the minimum $egmWin_0^-$ value at Y; then: (a) every pair of consecutive entries of $egmWinTab^-$ refers to a portion of $egmWin_R^-$ that is monotonically increasing or monotonically decreasing; (b) $egmWinTab^-$ covers enough of $egmWin_0^-$ that we are guaranteed that the solution $\bar{t}$ to $egmWin_0^-(\bar{t}) = egi$ lies between two consecutive entries of $egmWinTab^-$. Therefore the **while** loop in (59) is guaranteed to terminate, after which $\bar{t}$ is extracted by linear interpolation. The routine SOLVEMAX works in an entirely analogous way.

$$
\begin{array}{l}
\text{SOLVEMIN}(egi) \ = \\
\textbf{while} \ \neg[\ (egmWinTab^-(minIdx).val \le egi \le egmWinTab^-(minIdx+1).val) \lor \\
\qquad\qquad (egmWinTab^-(minIdx).val \ge egi \ge egmWinTab^-(minIdx+1).val)\ ] \\
\textbf{do} \ \textbf{if} \ minIdx < N_{MAX} - 1 \ \textbf{then} \ minIdx := minIdx + 1 \\
\quad \textbf{else} \ minIdx := 1 \\
\quad \textbf{fi} \\
\textbf{end} \ ; \\
\textbf{let} \ pretime = egmWinTab^-(minIdx).time \ , \\
\quad posttime = egmWinTab^-(minIdx+1).time \ , \\
\quad preval = egmWinTab^-(minIdx).val \ , \\
\quad postval = egmWinTab^-(minIdx+1).val \ , \\
\quad scale = (egi - preval)/(postval - preval) \ ; \\
\textbf{return(} \ pretime + scale \times (posttime - pretime) \,\textbf{)}
\end{array}
\tag{59}
$$

$$
\text{SOLVEMAX}(\ldots) \ = \ \ldots \tag{60}
$$

The working of CHECKMIN is relatively straightforward. It checks the new R value against the existing value $winRmin_A$, and then checks the recorded values of the electrogram-so-far corresponding to times between the parameters X and Y against what is permitted by $egmWin^-_R$ for the new R value. Routine CHECKMAX works entirely analogously.

$$\begin{array}{l}
\textsc{CheckMin}(Rval, \mathrm{X}, \mathrm{Y}) \ = \\
\textbf{if}\ \ Rval < winRmin_A\ \ \textbf{then}\ \ \textbf{return(}\ \mathsf{false}\ \textbf{)}\ \ \textbf{fi}\ \ \textbf{;} \\
\textbf{let}\ \ m = \lceil \mathrm{X}/\delta_t \rceil\ \ \textbf{;} \\
\textbf{while}\ \ m \in \mathrm{dom}\, egms_A \wedge \mathrm{X} < m\delta_t < \mathrm{Y} \\
\textbf{do}\ \ \textbf{if}\ \ egms_A(m) < egmWin^-_{Rval}(m\delta_t)\ \ \textbf{then}\ \ \textbf{return(}\ \mathsf{false}\ \textbf{)}\ \ \textbf{fi}\ \ \textbf{;} \\
\qquad m\ :=\ m+1 \\
\textbf{end}\ \ \textbf{;} \\
\textbf{return(}\ \mathsf{true}\ \textbf{)}
\end{array} \tag{61}$$

$$\textsc{CheckMax}(\ldots) \ = \ \ldots \tag{62}$$

6.12. On EGM$_{DVH}$ and EGM$_{AVH}$

The preceding completes our development of the sensing application. At the AVH level we have not been exhaustive, leaving certain details, such as the working of SOLVEMAX and CHECKMAX, and the quality of the interpolation tables $egmWinTab^-$ and $egmWinTab^+$, unspecified. Such facts would be needed to properly quantify the relationship between the DVH and AVH models.

We would normally like the AVH model to be a refinement of the DVH model. However, since $egmWin^-_R$ is not analytically invertible and we had to resort to an approximate method, the relationship between DVH and AVH is more complicated.

We argue as follows. By the design of the table $egmWinTab^-$, the linear interpolation approach aims to return a value very close to the exact inverse value of $egmWin^-_R$. For simplicity, let us suppose that the design is good enough for us to have an absolute bound Δ_{interp} on the difference between exact and interpolated inverse values for $egmWin^-_R$ (and for simplicity, let us suppose that the same bound also works for the maximum calculation). Then the DVH model and the AVH model values of $winRmin$ will be similarly close, except in the case that a failure to find a viable minimum value $winRmin$ has occurred in one model but not in the other, with this discrepancy being attributable to the small difference between exact and interpolated values. This situation will be signalled by one of $winRmin_B$ or $winRmin_A$ being equal to $R_{MAX,th}$ but not the other. Provided we 'avert our gaze' to this state of affairs, along the lines discussed earlier (doing analogously for the maximum calculation), then assuming we adjoin an equality relation on the inputs to the EGMD routine (which is the only other data we need), then we can set up a refinement between the DVH and the AVH models based on the following retrieve relation:

$$\begin{array}{l}
R_{DVH,AVH}(\langle egms_D, winRmin_D, winRmax_D, egirise_D, egifall_D\rangle, \\
\qquad\qquad \langle egms_A, winRmin_A, winRmax_A, egirise_A, egifall_A\rangle) \ \equiv \\
\quad egms_D = egms_A \wedge egirise_D = egirise_A \wedge egifall_D = egifall_A \ \wedge \\
\quad ((winRmin_D = R_{MAX,th} \Leftrightarrow winRmin_A = R_{MAX,th}) \Rightarrow |winRmin_D - winRmin_A| < \Delta_{\mathrm{interp}}) \ \wedge \\
\quad ((winRmax_D = R_{0,th} \Leftrightarrow winRmax_A = R_{0,th}) \Rightarrow |winRmax_D - winRmax_A| < \Delta_{\mathrm{interp}})
\end{array} \tag{63}$$

The main reason that a refinement based on $R_{DVH,AVH}$ works, is that the calculations for the minimum R values at the various time points $n\delta_t$ are largely independent (similar remarks apply for the maximum R values). A small error in the calculation at time $n\delta_t$ can become irrelevant when time $(n+1)\delta_t$ is considered. So errors are not compelled to accumulate. This is particularly good, since the accumulation of errors from state transition to state transition is a situation that usually spells the doom of any prospects of finding a useful refinement. Of course, it may be that one model finds itself in the non-viable case on the basis of a single time point calculation that, in absolute terms, is misguided, while the other model persists with viable minimum R values throughout. In such a case, the failing model will be stuck with its decision for the whole heartbeat. Such a situation, though less than ideal, is one from which our gaze is being averted by $R_{DVH,AVH}$.

7. LESSONS AND PRINCIPLES

Having completed our case study, it is time to draw out the general lessons it offers for the formal development of scientific applications in general, especially ones characterised by large elements of uncertainty. Here are our main conclusions.

[1] *Keep the modelling in the continuous sphere as long as possible, i.e. until the development is ready to be turned into an algorithm.* Above, we encountered many awkward technical issues to do with the behaviour of the input signal and the way it was being treated in our models. Such issues become considerably harder to deal with, and to relate to idealised models, when mixed up with the imprecision that accompanies discretization.

[2] *When dealing with the uncertainty coming from the observed environment, a range of plausible possibilities will need to be maintained*. Often this leads to a 'scoreboard-like' approach, as exemplified by our *winR* sets and their analogues.

[3] *The behaviour of many physical quantities is given by the solution of inverse problems, to which an approximate approach may be the best available.* We saw this in the need to invert the function $egmWin_R^-$ in our case study. Often, the physical law governing some behaviour may be straightforward to write down, but determining parameters such that the behaviour has desired characteristics, will require solving an inverse problem.

[4] *Solving inverse problems of the kind noted in **[3]** often leads to bifurcation and irregular behaviour of the solution set as parameters are varied.* We saw this in our discussion of the *winR* set as the peak of the spike in the electrogram was traversed. Such issues, the province of catastrophe theory [Saunders 1980; Gilmore 1981], arise surprisingly often in parameter variation situations because of the finite range of the physical parameters in genuine engineering applications. Techniques from numerical analysis can be expected to play an increasing role in formally developing such applications.

[5] *The ASM refinement semantics, in allowing (m,n) diagrams with $m,n > 1$, gives considerable help in tailoring refinement notions to fit the requirements of, and to also fit the inherent uncertainties of, complex scientific applications.* Such flexibility regarding refinement is very convenient in helping to 'avert the gaze' from portions of the development where rigorous statements cannot be made for reasons of complexity or of plain lack of knowledge of the true facts.

[6] *Adjust the nature of the proofs that accompany the various development steps to reflect the confidence with which they are believed to hold.* In our own development, much of the reasoning used in developing the case study was of a 'rigorous' rather than formal kind. We used insight about the behaviour of classes of real functions to inform our work, rather than deriving everything from first principles. (Such 'first principles' derivations could in any case be prohibitive.) In each application area, domain expertise has to determine what degree of effort is worthwhile regarding *ab initio* proof, and what can be captured in the axioms of the domain theory. The rule has to be, as always, that as one gets close to code, one has to be more and more precise, so that low level errors do not compromise the decisions made at higher levels. Machine support for varying levels of proof confidence would not be hard to arrange.

As is clear, these observations are of a rather general nature, so we can expect them to be relevant in the majority of scientific applications of a sufficiently complex nature.

8. RELATED TECHNICAL ISSUES

Our sensing development is typical of what are termed hybrid systems [Henzinger 1996; Alur and Dill 1994; He 1994; Alur et al. 1993; Carloni et al. 2006]. These, with the further addition of significant inter-processor communications to the systems architecture, are today referred to as cyber-physical systems [CPS 2008; National Science and Technology Council 2011; Sztipanovits 2011; Willems 2007; Zhang and He 2011; Zhühlke and Ollinger 2012; White et al. 2010; Barolli et al. 2011; Clarke and Zuliani 2011; Stehr et al. 2010; Lee 2006; 2008; Lee and Sesha 2013]. Both kinds of system are characterised by the unavoidable intrusion into the computing domain of physical law, and its critical dependence on time.

Our own development was characterised by a number of further technical choices. We chose ASM as the model based framework to work with, primarly for its very liberal refinement framework, based on $(m.n)$ diagrams, as noted in **[5]** above. We adapted this framework to yield a notion of the more flexible concept of retrenchment, for handling development steps for which refinement alone was not liberal enough. We used this to relate models whose timing properties were too different (A and CVH models) and later, to relate models where the precise discrepancy between the states at given moments was hard to calculate, though normally expected to be relatively small (B^*VH and DVH models).

In this latter case however, a notable feature of ASM refinement —as we described it in Section 4.2— is the absence of a commitment to any fixed idea about the relative progress of time in two refinement-related system models. This leads to the possibility of trading small variations in state against small variations in time to recover a refinement notion under suitably benign circumstances. This idea, in places referred to as retiming, has attracted some interest in the literature, and belongs to the wider approach of approximate simulation. Among relevant works we may cite [Stauner 2002; Doyen et al. 2005; Girard et al. 2008; Davoren 2009; Quesel et al. 2011; Tabuada 2009].

Our own choice to use retrenchment for such situations, instead of retiming (i.e. to only 'wiggle space a bit' rather than 'wiggle space *and* time a bit' in the language of [Davoren 2009]), was for one, admittedly subjective reason. The classical concept of universal time is so entrenched in human experience, including normal engineering situations, that we feel that it is dangerous to contemplate varying it without strong reasons. Allowing '2734665 milliseconds (after some commonly agreed starting point)' to mean different things in different models in a refinement chain, is potentially asking for trouble. It is sufficient to recall the Mars Lander, in which numerical values were inadvertently interpreted according to different units on the two sides of the Atlantic, to make the point.[6] Thus, although we prefer to interpret very similar mathematical facts via retrenchment, we concede that there is no *conceptual* barrier to the retiming approach. In the context of such an approach then, we would add the following reccommendation to our earlier list.

[7] *When using retiming techniques during refinement of successive system models, apply those retiming steps earliest, whose requirements permit the most liberal variations in the timing margins.* The risk in not doing so, is that a later development step may, through bringing in a timing variation more generous than those introduced earlier, introduce an overall variation in timing that breaks some requirement of an earlier development step.

9. CONCLUSIONS

In the preceding sections we briefly introduced some elements of heart functioning and heart block, and some basic elements of pacemakers, before setting up the time-domain pacemaker sensing problem as the target for our remaining investigations. We then outlined continuous versions of the ASM system modeling framework and the accompanying notions of refinement and retrenchment, so as to properly address the phenomena latent in our case study. This done, we plunged into a textbook version of the time-domain pacemaker sensing problem in the rest of the paper.

It is important to point out that what we did was indeed a textbook application. As we noted in the Introduction, 'what exactly are the requirements?' is a question which resonates loudly throughout the sphere of pacemaker design and development when one views it from a formal perspective. Ideally, now that the pacemaker world and the formal development world have come into contact, there is the opportunity for greater clarification to emerge round the requirements issue, but it will

[6] A notable exception to the 'classical time' observation is the GPS system, whose engineering required taking account of general relativistic corrections (including variations in time measures attributable to frames of reference under mutual acceleration), to achieve the projected accuracy. So variations in time can be forced by physics in classical engineering. Now imagine the additional complexity created by allowing the addition of further arbitrary variations in time to the mix, introduced to satisfy a notion of refinement.

always remain the case that the impetus for requirements must come from clinicians, even if formal developers might then lend such requirements a degree of precision that they have lacked hitherto.

Thus our study was based on a multitude of assumptions about the nature of pacemaker sensing in the clinical world which might not be justified in practice, since we ourselves are not clinicians. However, in our view, this does not detract from the value of our study. Once our assumptions were in place, they led us to confront technical challenges that would be found in many formal modelling and development applications characterised by large degrees of uncertainty.

Our development proceeded through various levels of abstraction. The key issue turned out to be the uncertainty regarding the point —during the notional heartbeat— at which the depolarization double pulse was deemed to actually take place (assuming that there was, indeed, a double pulse during the heartbeat). This caused the sensing process to have to estimate a range of possibilities, which became narrower and narrower as observations of the electrogram-so-far accumulated. Concomitant with this was the problem of transforming the abstract set of these possibilities into data that could actually be computed using conventional computational primitives. As we saw, this forced us to confront a multitude of technical issues of wide applicability.

The connections between the system models at the various pairs of adjacent levels of abstraction were mediated by refinements and retrenchments. Of course, being able to construct a refinement is the ideal, but we saw at a number of points that this ideal could not be attained. In such situations, retrenchment comes into its own, by providing a looser, yet still formal relationship between models.

From a practical perspective, it might reasonably be argued that the calculation-based sensing proposed in this paper could prove to be too power-hungry for clinical deployment. This is, first and foremost, a requirements issue such as we have pointed to in general terms above. Although the authors are not qualified to comment authoritatively on this point, the leading edge clinical literature on pacemakers (e.g. [Ellenbogen et al. 2006]), particularly when including discussion of the technology of implanted cardiac defibrilators, does allude to time domain based sensing approaches (while simultaneously underlining their compute-intensive and energy-intensive nature). Aside from its direct relevance to our concerns in this paper, this literature not only makes evident the considerable complexity of the pacing disorders that can affect the heart (and the difficulty of their correct diagnosis and the sometimes hazardous consequences of their incorrect diagnosis), but also rather spectacularly highlights the woeful inadequacy of [Boston Scientific 2007] as a representation of the true complexity of this domain.

Yet the landscape in regard of power and its management is ever-changing. Not only do we have devices today that could not have been dreamed of twenty years ago, but further developments continue apace. One can imagine that in the not too distant future, ultra-fast, ultra-low-power devices might become available, that could not be conceived today. For example, who can predict what developments based on graphene [Geim and Novoselov 2007; Britnell et al. 2012] might come about, that could completely change what can realistically be done with pacemakers and other low-power portable devices. Thus, not only could our study prove more applicable as technology advances, but as we have emphasised, the general techniques that we had to bring to bear on the problem have, in any case, much wider relevance.

REFERENCES

R. Alur, C. Courcoubetis, T. Henzinger, and P-H. Ho. 1993. Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems. In *Proc. Workshop on Theory of Hybrid Systems (LNCS)*, Vol. 736. Springer, 209–229.

R. Alur and D. Dill. 1994. A Theory of Timed Automata. *Theor. Comp. Sci.* 126 (1994), 183–235.

A. Aubert, B. Goldreyer, M. Wyman, E. Jaquemlyn, H. Ector, and H. de Geest. 1989. Filter Characteristics of the Atrial Sensing Circuit of a Rate Responsive Pacemaker. To See or Not to See. *PACE* 12 (1989), 525–536.

R. Banach and C. Jeske. 2014. Retrenchment and Refinement Interworking: the Tower Theorems. *Math. Struc. Comp. Sci,* (2014). To appear.

R. Banach, C. Jeske, and M. Poppleton. 2008. Composition Mechanisms for Retrenchment. *J. Log. Alg. Prog.* 75 (2008), 209–229.

R. Banach, M. Poppleton, C. Jeske, and S. Stepney. 2007. Engineering and Theoretical Underpinnings of Retrenchment. *Sci. Comp. Prog.* 67 (2007), 301–329.

R. Banach, H. Zhu, W. Su, and X. Wu. 2013. Moded and Continuous ASM. (2013). Submitted.

S. Barold, R. Stroobandt, and A. Sinnaeve. 2010. *Cardiac Pacemakers and Resynchronization Step by Step: An Illustrated Guide*. Wiley-Blackwell.

L. Barolli, M. Takizawa, and F. Hussain. 2011. Special Issue on Emerging Trends in Cyber-Physical Systems. *J. Amb. Intel. Hum. Comp.* 2 (2011), 249–250.

E. Börger. 2003. The ASM Refinement Method. *FACJ* 15 (2003), 237–257.

E. Börger and R.F. Stärk. 2003. *Abstract State Machines. A Method for High Level System Design and Analysis*. Springer.

Boston Scientific. 2007. PACEMAKER System Specification. (2007). http://www.cas.mcmaster.ca/sqrl/_SQRLDocuments/PACEMAKER.pdf.

L. Britnell, R. Gorbachev, R. Jalil, B. Belle, F. Schedin, A. Mishchenko, T. Georgiou, M. Katsnelson, L. Eaves, S. Morozov, N. Peres, J. Leist, A. Geim, K. Novoselov, and L. Ponomarenko. 2012. Field-Effect Tunneling Transistor Based on Vertical Graphene Heterostructures. *Science* (2012). DOI: 10.1126/science.1218461.

L. Carloni, R. Passerone, A. Pinto, and A. Sangiovanni-Vincentelli. 2006. Languages and Tools for Hybrid Systems Design. *Foundations and Trends in Electronic Design Automation* 1 (2006), 1–193.

E. Clarke and P. Zuliani. 2011. Statistical Model Checking for Cyber-Physical Systems. In *Proc. ATVA-11 (LNCS)*, Bultan and Hsiung (Eds.), Vol. 6996. Springer, 1–12.

CPS. 2008. *Report: Cyber-Physical Systems*. http://iccps2012.cse.wustl.edu/_doc/CPS_Summit_Report.pdf.

J. Davoren. 2009. Epsilon-Tubes and Generalized Skorokhod Metrics for Hybrid Paths Spaces. In *Proc. HSCC-09 (LNCS)*, Vol. 5469. Springer, 135–149.

V. DeCaprio, P. Hurzeler, and S. Furman. 1977. A Comparison of Unipolar and Bipolar Electrograms for Cardiac Pacemaker Sensing. *Circulation* 56 (1977), 750–755.

L. Doyen, T. Henzinger, and J.-F. Raskin. 2005. Automatic Rectangular Refinement of Affine Hybrid Systems. In *Proc. FORMATS-05 (LNCS)*, Vol. 3829. Springer, 144–161.

K. Ellenbogen, B. Wilkoff, G. Kay, and C-P. Lau. 2006. *Clinical Cardiac Pacing, Defibrillation and Resynchronization Therapy*. Saunders.

K. Ellenbogen and M. Wood. 2008. *Cardiac Pacing and ICDs*. Wiley-Blackwell. 5th ed.

S. Furman, P. Hurzeler, and V. DeCaprio. 1977. Cardiac Pacing and Pacemakers III: Sensing the Cardiac Electrogram. *Am. Heart J.* 93 (1977), 794–801.

A. Geim and K. Novoselov. 2007. The Rise of Graphene. *Nature Materials* 6 (2007), 183–191.

R. Gilmore. 1981. *Catastrophe Theory for Scientists and Engineers*. Dover.

A. Girard, A. Julius, and G. Pappas. 2008. Approximate Simulation Relations for Hybrid Systems. *Disc. Event Dyn. Syst.* 18 (2008), 163–179.

B. Goldman, E Noble, J. Heller, and D. Covvey. 1974. The Pacemaker Challenge. *Can. Med. Ass. J.* 10 (1974), 28–31.

A. Gomes and M Oliveira. 2009. Formal Specification of a Cardiac Pacing System. In *Proc. FM-09 (LNCS)*, Vol. 5850. Springer, 692–707.

A. Gomes and M Oliveira. 2010. Formal Development of Cardiac Pacemaker: From Specification to Code. In *Proc. SBMF-10 (LNCS)*, Vol. 6527. Springer, 210–225.

J. He. 1994. From CSP to Hybrid Systems. In *A Classical Mind, Essays in Honour of C.A.R. Hoare*, Roscoe (Ed.). Prentice-Hall, 171–189.

T. Henzinger. 1996. The Theory of Hybrid Automata. In *Proc. IEEE LICS-96*. IEEE, 278–292. Also http://mtc.epfl.ch/~tah/Publications/the_theory_of_hybrid_automata.pdf.

C.B. Jones, P. O'Hearne, and J. Woodcock. 2006. Verified Software: A Grand Challenge. *IEEE Computer* 39 (2006), 93–95.

M. Keinert, H. Elmqvist, and H. Strandberg. 1979. Spectral Properties of Atrial and Ventricular Endocardial Signals. *PACE* 2 (1979), 11–19.

KIV*Karlsruhe Interactive Verifier*. http://www.informatik.uni-augsburg.de/lehrstuehle/swt/se/kiv/.

F. Kusumoto and N. Goldschlager. 2007. *Cardiac Pacing for the Clinician*. Springer.

E. Lee. 2006. Cyber-Physical Systems — Are Computing Foundations Adequate. In *NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*.

E. Lee. 2008. Cyber-Physical Systems: Design Challenges. In *Proc. ISORC-08*. Invited Paper. http://chess.eecs.berkeley.edu/pubs/427.html.

E. Lee and S. Sesha. 2013. *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*. Lulu.com.

H. Macedo, P. Larsen, and J. Fitzgerald. 2008. Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System Using VDM. In *Proc. FM-08 (LNCS)*, Vol. 5014. Springer, 181–197.

D. Méry and N. Singh. 2009. *Pacemaker's Functional Behaviors in Event-B*. Technical Report INRIA-00419973:2. LORIA, Université Henri Poincaré - Nancy I. http://hal.inria.fr/inria-00419973/PDF/Pacemaker.pdf.

D. Méry and N. Singh. 2010. *Technical Report on Formal Development of Two-Electrode Cardiac Pacing System*. Technical Report INRIA-00465061:2. LORIA, Université Henri Poincaré - Nancy I. http://hal.inria.fr/inria-00465061/PDF/Report_2electrode.pdf.

D. Méry and N. Singh. 2011. *Functional Behavior of a Cardiac Pacing System*. Technical Report. LORIA, Université Henri Poincaré - Nancy I. http://www.loria.fr/~singhnne/Home_files/ downloads/ijdecs2010.pdf, Int. J. Discrete Event Control Systems.

National Science and Technology Council. 2011. Trustworthy Cyberspace: Strategic plan for the Federal Cybersecurity Research and Development Program. (2011). http://www.whitehouse.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf.

A. Platzer. 2010. *Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics*. Springer.

J.-D. Quesel, M. Fr anzle, and W. Damm. 2011. Crossing the Bridge Between Similar Games. In *Proc. FORMATS-11 (LNCS)*, Vol. 6919. Springer, 160–176.

RET*Retrenchment Homepage*. http://www.cs.man.ac.uk/retrenchment.

P. Saunders. 1980. *An Introduction to Catastrophe Theory*. Cambridge University Press.

G. Schellhorn. 2001. Verification of ASM Refinements Using Generalized Forward Simulation. *JUCS* 7 (2001), 952–979.

G. Schellhorn. 2005. ASM Refinement and Generalizations of Forward Simulation in Data Refinement: A Comparison. *Theor. Comp. Sci.* 336 (2005), 403–435.

A. Schuchert, A. Aydin, C. Israel, G. Gaby, and V. Paul. 2005. Atrial Pacing and Sensing Characteristics in Heart Failure Patients Undergoing Cardiac Resynchronization Therapy. *Europace* 7 (2005), 165–169.

T. Stauner. 2002. Discrete-Time Refinement of Hybrid Automata. In *Proc. HSCC-02 (LNCS)*, Vol. 2289. Springer, 144–161.

M. Stehr, M. Kim, and C. Talcott. 2010. Toward Distributed Declarative Control of Networked Cyber-Physical Systems. In *Proc. UIC-10 (LNCS)*, Yu, Liscano, Chen, Zhang, and Zhou (Eds.), Vol. 6406. Springer, 397–413.

J. Sztipanovits. 2011. Model Integration and Cyber Physical Systems: A Semantics Perspective. In *Proc. FM-11*, Butler and Schulte (Eds.). Springer, LNCS 6664, p.1, http://sites.lero.ie/download.aspx?f=Sztipanovits-Keynote.pdf. Invited talk, FM 2011, Limerick, Ireland.

P. Tabuada. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer.

J. White, S. Clarke, C. Groba, B. Dougherty, C. Thompson, and D. Schmidt. 2010. R&D Challenges and Solutions for Mobile Cyber-Physical Applications and Supporting Internet Services. *J. Internet Serv. Appl.* 1 (2010), 45–56.

J. Willems. 2007. Open Dynamical Systems: Their Aims and their Origins. Ruberti Lecture, Rome. (2007). http://homes.esat.kuleuven.be/~jwillems/Lectures/2007/Rubertilecture.pdf.

F. Wilson, A. Macleod, and P. Barker. 1933a. The Distribution of the Action Currents Produced by Heart Muscle and Other Excitable Tissues Immersed in Extensive Conducting Media. *J. Gen. Physiol.* 16 (1933), 423–456.

F. Wilson, A. Macleod, and P. Barker. 1933b. *The Distribution of the Currents of Action and of Injury Displayed by Heart Muscle and Other Excitable Tissues*. University of Michigan Studies. Scientific Series, Vol. 10. Ann Arbor: University of Michigan Press. Reprinted in: Lepeschkin and Johnston (eds.), Selected Papers of Frank N. Wilson. Ann Arbor, J.W. Edwards (1954).

J. Woodcock. 2006. First Steps in the The Verified Software Grand Challenge. *IEEE Computer* 39 (2006), 57–64.

J. Woodcock and R. Banach. 2007. The Verification Grand Challenge. *JUCS* 13 (2007), 661–668.

L. Zhang and J. He. 2011. A Formal Framework for Aspect-Oriented Specification of Cyber Physical Systems. In *Proc. ICHIT-11 (CCIS)*, Lee, Howard, and Slezak (Eds.), Vol. 206. Springer, 391–398.

L. Zhühlke and L. Ollinger. 2012. Agile Automaton Sysytems Based on Cyber-Physical Systems and Service Oriented Architectures. In *Proc. ICAR-12 (LNEE)*, Lee (Ed.), Vol. 122. Springer, 567–574.