Research Proposal:
A Specification Language for Emergent Properties

PhD Candidate: Alena Rodionova
Supervisor: Prof. Radu Grosu
Co-supervisors: Prof. Agata Ciabattoni,
Asst. Prof. Ezio Bartocci

Institute of Information Systems, Vienna University of Technology
alena.rodionova@tuwien.ac.at

Contents

1 Introduction

The formal specification of emergent behaviour is a great challenge: first, the logic has to be able to capture temporal properties, which manifest themselves both in the time and in the (dual) frequency domains. Second, the logic has to be able to capture complex spatial properties. Third, the logic has to be robust to analog switching, as most physical and biological processes "take decisions" this way, and therefore a quantitative fuzzy version is very desirable. Fourth, the logic should not only be expressive enough, but it ought also to have good decision-theoretic properties, thus the complexity question of such logic should be exposed to the detailed analysis.

Some of the most important and intriguing research questions in the rapidly expanding domains of cyber-physical (CPS) and biological (BS) systems, are the following two questions:

1. How does one formally specify the emergent behaviour?
2. How does one efficiently predict and detect its onset?

An emergent behaviour is a global, qualitatively novel behaviour of a system, which results from the interaction of the system's components, is maintained for a specific period of time, and ceases to exist when particular components are removed from the system. The formation of a complex snowflake patterns, the build-up of a traffic jams, the erection of termite cathedrals, the conditions leading to cancer, whirlpool-like spirals in the heart's electrical activity and various complex medical devices such as Implantable Cardioverter Defibrillators (ICDs) or Pacemakers are prominent BS and CPS examples.

The automatic analysis of emergent behaviour in CPS and BS is a great challenge: nonlinear dynamics, mixed discrete-continuous behaviour, stochasticity and fuzziness, are features that may render the analysis intractable. Moreover, these processes are usually hierarchically aggregated. For example, the human heart, has to be captured from molecular and cellular, to tissue and to organ level, and defects at any level may impact its emergent behaviour.

Emergent properties are similar to other spatio-temporal patterns such as air and water turbulences, geological movements and sound vibrations. All these domains have developed their own ways of characterizing important emergent situations. In the case of turbulence in liquid flow, for example, there are regions in phase space, that separate the different worlds of laminar and turbulent flows. Similarly, we expect to find different regions in the CPS and BS phase space, corresponding to different types of stable and unstable dynamics.

As a result, we rely on these multi-disciplinary source cases listed above as an inspiration to define the formalism most suitable for the logical specification of the emergent phenomena in CPS.

2 State of the Art

Cyber-Physical Systems (CPS) - are engineered systems that are built from, and depend upon, the integration of computational algorithms and physical components. CPS collaborate computational elements controlling physical entities. On the one hand, digital systems operate in a discrete manner, where computation and communication proceed in synchronization with the processor cycles. On the other hand, physical systems execute continuously in dense real-time. Thus CPS - are complex systems exhibiting both *discrete* and *continuous* behaviours. Moreover, in such systems *time* domain information also interacts with *space* domain, *energy*, and *uncertainty*.

Any CPS study consists of the following aspects of the system behaviour:

- specification (e.g. spatial-temporal logics);
- design (e.g. probabilistic hybrid systems);
- analysis (e.g. symbolic and stochastic model checking;
- control (e.g. PID, supervisory, and optimal).

2.1 Temporal Reasoning

Discussion of reasoning about time goes back to ancient philosophy. Starting from Zeno's paradoxical arguments [1] refer to the nature of time and the question of infinite divisibility of time intervals, people are trying to analyse and formalise such arguments in order to understand the nature of time.

Nowadays, the crucial point in analysis of any reactive system is the reasoning about ongoing input-output behaviour of a system, rather than just the final output of a complex program. *Temporal logic* (TL) provides a convenient framework for writing specifications that the system under design should satisfy in a compact and formal way. In short, the main idea of TL is that a formula is not statically true or false in a model, as it is in propositional and predicate logic. Instead, the models of TL contain several states and a formula can be true in some states and false in others.

It was initially intended to evaluate clean and well-defined sequences of states and events as found in digital systems. The most widely used logic for specifying the behaviour of such systems is a linear temporal logic (LTL) [16].

Metric temporal logic (MTL)(see [11] for more details) is a real-time extension of LTL. For the purpose of verification, it may contain both *future* and *past* temporal operators. The advantage of such operators is that they can be naturally monitored online, which is particularly important for monitoring real systems. Its principal modalities are called timed until $\mathcal{U}_I$ and timed since $\mathcal{S}_I$.

Formally, the syntax of MTL is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 U_I \varphi_2 \mid \varphi_1 S_I \varphi_2,$$

where $p \in P$ is an atomic proposition and I is a non-empty interval of the form $\langle a, b \rangle$, such that, the left boundary $\langle$ is either open (or closed [, the right boundary $\rangle$ is either open $)$ or closed $]$, and the boundary values $a, b \in \mathbb{N}$ are natural numbers with $0 \leq a \leq b$.

The satisfaction of a given formula φ with respect to a Boolean signal $x : \mathbb{T} \to 2^P$ at time point i is a relation denoted by $(x, i) \models \varphi$ and is defined inductively with the classical rules for atomic propositions and Boolean operators, and with the following rules for $\mathcal{U}_I$ and $\mathcal{S}_I$ modalities:

$$
\begin{aligned}
(x, i) \models \varphi \mathcal{U}_I \psi \quad &\iff \quad \exists j \in (i + I) \cap \mathbb{T} : (x, j) \models \psi \\
& \qquad \text{and } \forall k \in (i,\, j),\ (x, k) \models \varphi \\
(x, i) \models \varphi \mathcal{S}_I \psi \quad &\iff \quad \exists j \in (i - I) \cap \mathbb{T} : (x, j) \models \psi \\
& \qquad \text{and } \forall k \in (j,\, i),\ (x, k) \models \varphi.
\end{aligned}
$$

From the basic definition of MTL, one can derive other standard Boolean $(\top = p \vee \neg p, \quad \varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi))$ and temporal operators:

$$
\begin{aligned}
\Diamond_I \varphi = \top \mathcal{U}_I \varphi, \quad & \Box_I \varphi = \neg \Diamond_I \neg\varphi, \\
\diamondminus_I \varphi = \top \mathcal{S}_I \varphi, \quad & \boxminus_I \varphi = \neg \diamondminus_I \neg\varphi.
\end{aligned}
$$

It should be mentioned, that the Finally $\Diamond_I$, Globally $\Box_I$, Once $\diamondminus_I$ and Historically $\boxminus_I$ operators also admit a natural direct definition of their semantics. For instance, future operators $\Diamond_I$ and $\Box_I$ can be defined as follows:

$$
\begin{aligned}
(x, i) \models \Diamond_I \varphi \quad &\iff \quad \exists j \in (i + I) \cap \mathbb{T} : (x, j) \models \varphi \\
(x, i) \models \Box_I \varphi \quad &\iff \quad \forall j \in (i + I) \cap \mathbb{T}, (x, j) \models \varphi
\end{aligned}
$$

Though MTL was proven to be extremely fruitful in verifying hardware [7, 15], in the last couple of years, it had been extended to be able to specify properties of real-valued signals defined over dense time. Logic over such signals is called STL.

Signal temporal logic (STL) [13, 12] allows designers to speak of properties related to the order of discrete events and the temporal distance between them, where "events" correspond to changes in the satisfaction of some predicate (e.g., threshold crossing) over the real variables [6]. One of the technical impediments to the adoption of STL, is its purely *time-domain nature*; i.e., it does not lend itself to frequency-domain analysis. This kind of analysis is based on the Fourier spectrum of the signal which in many engineering applications is more important than the properties of the signal itself (See Section 2.2). There exist few important reasons of exhibiting such interest in frequency domain features capturing.

- First, real-life analog signals are accompanied by omnipresent noise, i.e., random perturbations of the desired signal. Extraction of such signal component is possible during working with an initial signal in the frequency domain. Such operation in Signal Processing is known as *filtering*. Typically, the noise component of a signal populates a range of frequencies different from the range of the signal of interest.

- Second, an analog signal is usually a composition of multiple sources, and one of the analysis steps is the separation of these sources. In the frequency domain, this is done by simple operations such as *thresholding* or *filtering the range of frequencies* of interest, assuming that each source has a range clearly distinct from the ranges of others.

- Third, for some situations it is more important to know the frequency component of a signal rather than timing information. It is possible to reconstruct a signal dealing only with its frequency range, but for instance, the information about its time duration is not essential and can be neglected from analysis.

Moreover, it should be mentioned that real-time logics such as MTL and STL allow to assign not only a standard Boolean semantics, but also a quantitative one. The *quantitative semantics* defines a function mapping a property φ and a signal x to a real number, known as the *robust satisfaction value*. A large positive value suggests that x easily satisfies φ, a positive value near zero suggests that x is close to violating φ.

2.2 Frequency Analysis

The essence of frequency analysis is that any signal can be transformed into an alternative representation consisting of a weighted sum of basic elementary signals. Most of the time, such basis is a set of sinusoids of various frequencies and phases.

Thus, the signal under analysis is transformed on any interval $[0, T_0]$ from a time-domain representation $x : \mathbb{T} \to \mathbb{R}$ to a function $\hat{x}$ mapping frequencies to their coefficients which is called *Fourier series*:

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos(2\pi\omega_k t) + b_k \sin(2\pi\omega_k t). \tag{1}$$

This can be written more concisely using *Euler's formula* $e^{ix} = \cos(x) + i \sin(x)$:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k \, e^{2\pi i\omega_k t}, \tag{2}$$

where coefficients c_k provide a discrete spectrum for a signal x on $[0, T_0]$. By contrast, the *Fourier Transform* maps x on the whole time domain to a continuous spectrum $\{c_\omega : \omega \in \mathbb{R}\}$ containing all real frequencies. The *Inverse Fourier Transform* (IFT), which recovers x from its spectrum, can be written as:

$$x(t) = \int_{-\infty}^{+\infty} c_\omega \, e^{2\pi i\omega t} \, \mathrm{d}\omega, \qquad c_\omega = \hat{x}(\omega) = \int_{-\infty}^{+\infty} x(t) \, e^{-2\pi i\omega t} \, \mathrm{d}t, \tag{3}$$

which can be seen as a generalization of the Fourier series equation (2).

One of the extensions of classical FT method is called *the short-time* or *windowed Fourier transform* (STFT), which is based on a product of initial signal with a window function, whose purpose is simply to filter the values of signal outside a neighbourhood of the small interval by forcing them to be 0. Consequently, the STFT of the signal x defines a two-dimensional spectrum $\{c_{\omega,\tau} : (\omega,\tau) \in \mathbb{R}^2\}$. The STFT can be visualized as a *spectrogram*, which plots the norms of the coefficients $c_{\omega,\tau}$ as a surface above the time-frequency plane (ω,τ).

2.3 Formal Verification of ICDs

Implantable Cardioverter Defibrillators (ICDs) are medical devices at the forefront of preventing sudden death in patients suffering from Ventricular Arrhythmias. ICDs detect anomalous patterns of electrical excitation in the ventricles (lower chambers) and atria (upper chambers) of the heart and then supply appropriate life-saving electrical shocks to restore the normal sinus rhythm. Two such anomalous patterns are Ventricular Tachycardia (VT) and Ventricular Fibrillation (VFib). VT corresponds to rapid beating of the ventricular tissue and may lead to VFib, which is potentially fatal, and is characterized by uncontrollable quivering of the heart muscle. Extensive studies [2], have shown that ICD implantation significantly reduces mortality by reducing the risk of sudden death in patients suffering from ventricular dysfunction and related arrhythmias.

Due to sophisticated algorithms, ICDs have evolved into complex CPS that use embedded software and hardware to diagnose arrhythmias based on the electrograms (EGMs) sensed by the leads, and control the heart under varying physical conditions.

A major safety concern for transvenous ICDs is minimizing Inappropriate Shocking (IS) [17], which occurs when unnecessary shocks are delivered either due to sensing errors, or misdetection of arrhythmias. IS has short-term adverse effects such as extreme pain and anxiety, as well as long-term consequences, such as depression and different types of stress disorder. Reliability requirements include maintaining the heart rate and atrio-ventricular rate ratios within nominal ranges.

Verification and Validation (V&V) of the above-mentioned requirements is critical. Formal verification techniques can be thought of as comprising three parts [9]:

- a *framework for modelling systems*, typically a description language.
- a *specification language* for describing the properties to be verified.
- a *verification method* to establish whether the description of a system satisfies the specification.

Thus to verify that a system satisfies a property we must do three steps:

- model the system using the description language of a model checker, arriving in a model $\mathcal{M}$.

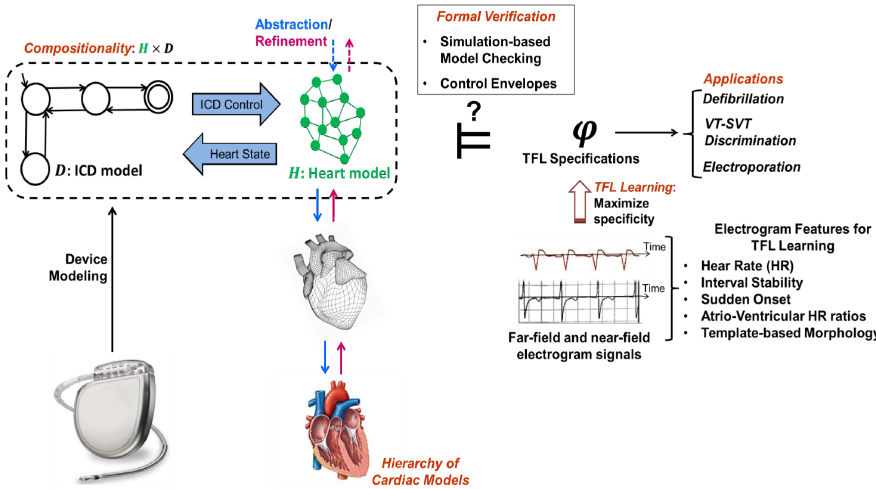

Figure 1: Verification Framework for ICDs

- code the property using the specification language of the model checker, resulting in a temporal logic formula ϕ.
- run the model checker with inputs $\mathcal{M}$ and ϕ.

The model checker outputs the answer 'yes' if $\mathcal{M} \models \phi$ and 'no' otherwise.

The Figure 1 shows the detailed verification framework for ICDs. It consists of the abstraction of the heart model H and the ICD model D. Verification process asks whether a composed model $H \times D$ satisfies ϕ:

$$H \times D \models \phi, \tag{4}$$

where ϕ represents the property to be verified. TL-based requirements for the various applications shown in the figure will be learned from electrograms. The learning process will emphasize the specificity of the TL formulae to enable better VT/SVT discrimination.

3 Present and Future Work

3.1 Subgoals and Methodology

There are four major immediate challenges to tackle:

- to combine event-based time-domain properties with frequency domain, using shifting window transforms such as Wavelets or STFT. In preliminary results [3], authors enriched the Signal Temporal Logic operators with frequency predicates and have used this logic to monitor with success music patterns like blues. We plan to extend this logic with more flexible and robust frequency operators, and to use it for formal specification of patterns that are typical, for example, to cell-, tissue- and full heart malfunctioning, like the electrocardiogram (ECG) characteristic of the Brugada syndrome.
- to associate ideas from modal logic and pattern recognition to come up with an useful language for defining spatial forms. Our work will be based on the ideas from the paper of Grosu et al. [8]. In this paper authors developed a linear spatial-superposition logic (LSSL) where, for instance, the next operator X is a spatial, zoom-in operator.
- to integrate the above for an effective logical formalism encompassing both time and space in order to explore the behaviour of complex CPS.
- to develop an algorithm for automatic checking whether a system will exhibit a given emergent behaviour, and also infer the parameter ranges for which it does.

3.2 Research Questions

Although the research challenges, discussed in Section 3.1 are of great interest for us, here I provide our main research lines and tasks that we are currently working on.

- The 1st task is to study the various formalisms used to classify spatially-extended phenomena, in dynamic image recognition, signal processing and medicine as well as existing spatial logics.
- The 2nd task is to extend temporal logic with frequency domain properties based on dynamic (shifting window) versions of the Fourier transform and devise a monitoring mechanism for properties expressed in this logic.
- The 3rd task is to develop a formalism to describe still pictures based on a combination of ideas from spatial modal logic and pattern recognition theory.
- The 4th task is to combine the ideas described above to develop a formalism that combines space, time and frequency domains.
- Finally, the 5th task is to develop algorithms for supervised learning spatio-temporal specifications of certain emergent properties in CPS.

3.3 Collaboration

In classical temporal logics, patterns are characterized with predicates that partition the set of values into two subsets: the one that satisfies the pattern and the other one that does not. This kind of predicates however, are usually difficult to learn automatically or even to manually determine, because the threshold values distinguishing these two sets are usually fuzzy.

The collaboration with Professor Ciabattoni and Professor Baaz who are experts in multi-valued logics such as Goedel and Lukasiewicz logics, will help us to investigate the appropriate specification formalism for our case studies.

Similarly, the possible future collaboration with Professor Veith, Professor Bloem and Professor Biere, will help us in the temporal domain.

Collaborators within the DK are :

- Helmut Veith, an expert in compositional methods for temporal logic verification.
- Armin Biere, an expert in SAT/SMT-based model checking techniques.
- Agata Ciabattoni, an expert in fuzzy logic theory, databases and AI.
- Matthias Baaz an expert in multivalued logics.

Although we have not yet collaborated with any of these investigators before, the current proposal will be a very important catalyst for a successful collaboration in the future.

4 Progress

In this section I will report on my progress during the first year of my doctoral studies.

4.1 Temporal Logic as Filtering

The following section is based on our recent accepted paper to the 19th International Conference HSCC 2016: *Temporal Logic as Filtering* [18].

In this paper we have shown that MTL (see Section 2.1) can be viewed as linear time-invariant filtering, by interpreting addition, multiplication, and their neutral elements, over the idempotent dioid (max,min,0,1). Moreover, by interpreting these operators over the field of reals (+,×,0,1), one can associate various quantitative semantics to a metric-temporal-logic formula, depending on the filter's kernel used: square, rounded-square, Gaussian, low-pass, band-pass, or high-pass.

In order to illustrate the effect of applying various smooth kernels to a signal x_p, consider the example shown in Figure 2. In Figure 2(A) we show the signal x_p. In Figure 2(B) we illustrate the quantitative semantics with respect to a square kernel. In Figure 2(C) we show the same semantics with respect to a sigmoidal window. This automatically adds tolerance to a time-jitter at the boundaries of the kernel. Finally, in Figures 2(D-E) we show the result of applying Gaussian kernels, with different values of reciprocal of standard deviation.

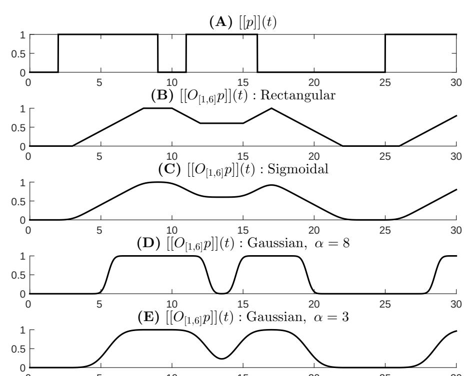

Figure 2: Smooth evaluation of $\llbracket \diamondminus_{[1,6]}\, p \rrbracket\,(t)$.

This remarkable connection between filtering and metric temporal logic allows us to freely navigate between the two, and **to regard signal-feature detection as logical inference**. To the best of our knowledge, this connection has not been established before. We prove that our qualitative, filtering semantics is identical to the classical MTL semantics. We also provide a quantitative semantics for MTL, which measures *the normalized, maximum number of times a formula is satisfied within its associated kernel, by a given signal.*

Example 1. Consider the discrete-time signal x over the interval $[0, 12]$, with $\Delta t = 1$, shown in Figure 3. Since it is discrete-time, its domain is $\{0, 1, \ldots, 12\}$. In order to interpret the MTL operator $\diamondminus_{[1,4]}\, p$, we construct the rectangular windowing function $w^{-}_{[1,4]}[i]$. Since the formula $\diamondminus_{[1,4]}\, p$ is a past MTL formula,

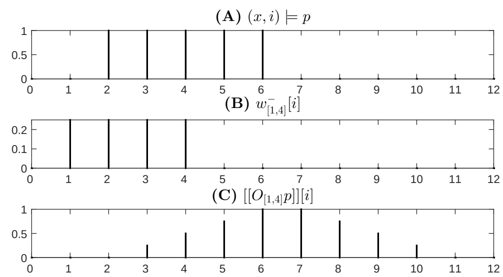

Figure 3: Evaluation of $[\![\diamondminus_{[1,4]}\, p]\!](t)$

MTL Formula	Time point			
	3	4	5	6
$[\![\diamondminus_{[1,4]}\, p]\!]$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1

Table 1: Quantitative semantics values

the corresponding window is formally defined as follows:

$$w^{-}_{[1,4]}[i] = \frac{1}{4}\sum_{j=1}^{4}\delta(i-j). \tag{5}$$

The representation of this window is shown in Figure 3. Now we are able to evaluate the quantitative semantics of signal x with respect to the Once-formula given below:

$$[\![\diamondminus_{[1,4]}\, p]\!]\,[i] = \sum_{j=0}^{12} p[j]\cdot w^{-}_{[1,4]}[i-j].$$

The quantitative-values of the result are given in a Table 1, and their graphical representation is shown in Figure 3.

We show that this semantics is sound, in the sense that, it produces a measure greater than 0, if and only if, the formula is satisfied by the discrete-time qualitative semantics, and 0, otherwise:

Theorem 1 (Soundness)**.** *For a positive normal form MTL formula φ and a discrete-time signal x, it holds that x satisfies φ, if and only if, its quantitative semantics is strictly greater than 0. It does not satisfy φ, if and only if, its quantitative semantics is 0:*

$$(x,i)\models\varphi \iff [\![x,\varphi]\!][i] > 0,$$
$$(x,i)\not\models\varphi \iff [\![x,\varphi]\!][i] = 0.$$

We have implemented both of our semantics in Matlab, and illustrate their properties on various formulas and signals, by plotting their computed measures.

4.2 Case Study: Music Analysis. Note Detection

The following section is based on the paper of Donze et al. (2012) [5], where authors have presented a new specification formalism for real-valued signals that

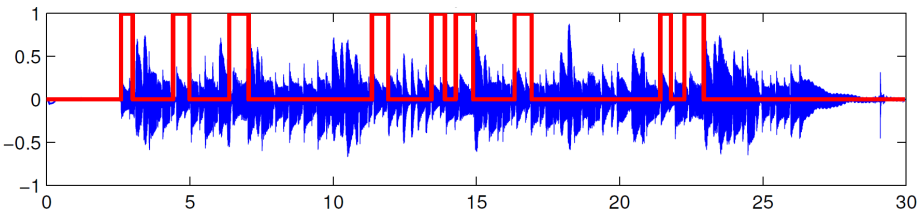

Figure 4: Recorded Blues melody and satisfaction of μ_E

combines temporal logic properties in the time domain with frequency-domain. Such logical formalism allows to express more complex signal patterns which are defined not only in time domain, but also in frequency domain.

By considering the frequency as a parameter, it is possible to obtain a family of signal operators $\{f_{L,\omega}\}$, such that $y = f_{L,\omega}(x)$, if $y(t) = \hat{x}_L(\omega, t)$, where x is a signal and L is a length of a window function of STFT. In other words, $f_{L,\omega}$ is a projection of the spectrogram of x on the ω frequency.

Novel *time-frequency logic* (TFL) is obtained by adding the operators $f_{L,\omega}$ to STL (see Section 2.1 for details). A spectral signal $y = f_{L,\omega}(x)$, like any other signal, can participate in any TFL formula as an argument of predicates. TFL allows to define properties of musical melodies and monitor different acoustic waveforms starting with the basic task of note detection.

Since each musical note is basically a sinusoid oscillating at a specific frequency value or *pitch*, note detection predicate can defined a spectral STFT operator:

$$\text{pitch}_{\omega,x}(t) = \hat{x}_L(\omega, t), \tag{6}$$

which is the amplitude of the frequency ω in a signal x around time t. Such operator must be able to discriminate a note from its two closest neighbouring notes, which have frequencies $\omega_1 = 2^{-\frac{1}{12}}\omega$ and $\omega_2 = 2^{\frac{1}{12}}\omega$, so for instance, note E detecting predicate in TFL can be defined as a

$$\mu_E \;=\; \text{pitch}_{\omega_E}(x) > \theta, \quad \text{where} \;\; \omega_E = 329 \text{ Hz}. \tag{7}$$

The result is presented in Figure 4. In order to illustrate the expressiveness of TFL, let us consider an example of simple Blues melody recognition: the framework has been tested by trying to verify that a guitar melody, played and recorded by one of the authors, was indeed a Blues melody. TFL specification in such case is defined as follows:

$$\varphi_{\text{blues}} = \mu_E \wedge \Diamond_{[5b,\,6b]}\, \mu_A \wedge \Diamond_{[8b,\,9b]} \left(\mu_B \wedge \Diamond_{[b,\,2b]}\, \mu_A \wedge \Diamond_{[2b,\,3b]}\, \mu_E\right) \tag{8}$$

where b is a duration of one bar. Moreover, TFL specification formalism described in this section is supported by a full monitoring algorithm implemented in the Breach tool [4], and tested on real acoustic signals.

First extension which has already been done by us, includes the use of more versatile Wavelet Transform for time-frequency analysis of an input musical signal. And second extension is based on the adaptation of pitch detection of a signal by Specmurt method [19].

4.3 Case Study: EGM Analysis. VT/SVT Discrimination

The following section is based on the ongoing work within the CyberCardia project, which is a part of the NSF's centre-scale initiative to advance the state-of-the-art in CPS.

ICDs are expected to *reliably* maintain the heart rate within a pre-specified interval. A major *safety* concern with ICD is an inappropriate shocking, which is defined as the delivery of inappropriate or unnecessary electrical shocks either due to sensing errors, or misdetection of VT on the basis of electrogram signals sensed by the leads of the ICD. SVT, such as AFib, originate in the upper chambers of the heart and ICD shocks must be ideally *withheld in such cases* (see Section 2.3).

Since electrograms form the basis of VT-SVT discrimination, requirements will be specified using temporal and frequency-domain aspects of electrograms. Following are some of the features of electrograms that will be used to specify the requirements:

- *Heart Rate* has an inverse relationship with the cycle length, which is defined as the time interval between two consecutive R peaks of the periodic electrogram.
- *Stability* refers to the variability in cycle lengths. VT is characterized by reasonably stable cycle lengths, whereas SVT, such as AFib, shows higher beat-to-beat variability.
- *Sudden Onset* criterion compares the $R - R$ interval to mean values and measures any steep changes in the cycle lengths, which indicate sinus tachycardia.
- *QRS Complex Morphology* compares the shape of the QRS complex to nominal patterns measured a priori under sinus rhythm conditions. Medtronic's ICDs use the Wavelet operation, which involves monitoring windows of eight successive QRS complexes and matching them with nominal templates.
- *Atrio-Ventricular Heart Rate Ratios.* Dual-chamber ICDs have two leads that sense electrograms from both the ventricles and the atria. For such ICDs, the ratio between the measured heart rates from the two regions is the primary VT-SVT discriminant.

Such methodology is used by *Medtronic Discrimination Algorithm* [10, 14].

A Time-Frequency Logic (TFL) will be used to capture these features. Based on these features, the semantics of the logic will allow us to define VT and SVT and use the definitions for verifying the ICDs. Such logic is based on STL (see 2.1 for details) and Frequency analysis techniques (see 2.2).

The decision tree of Medtronic dual chamber algorithm is presented on a Figure 6.

Heart rate and Stability

The cycle length (CL) or heart rate (HR) is a fundament in the detection of tachycardia in ICD patients. A sustained ventricular HR in adults > 250 bpm or a CL < 250 ms is very specific for fast VT or VF (Figure 5 red zone).

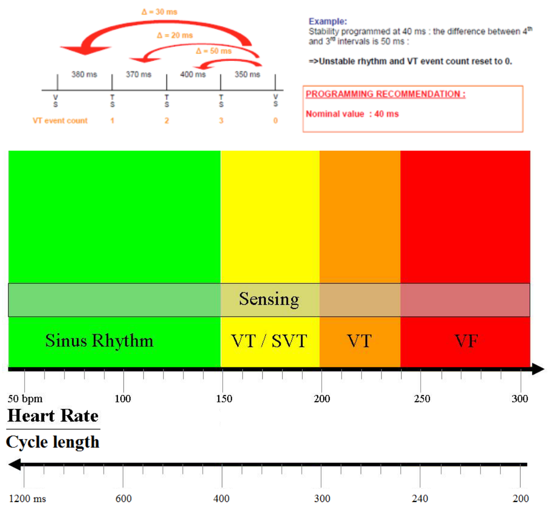

Figure 5: Stability and Heart rate

VT is primarily characterized by an increase in the heart rate. Stability criteria intervenes when the VT event count reaches 3 (see Figure 5 for details). An interval is classified as unstable if its length varies from the 3 previous intervals by a difference greater than the programmed stability interval. If the device detects an unstable interval, SVT is detected and VT event counter is reset to 0.

Such properties can be specified by using STL, a real-valued cycle-length input signal ξ and a set of defined threshold values as follows:

$$
\begin{aligned}
(\xi, t) \models \text{“VT/FVT/VF zone”} \quad &\Longleftrightarrow \quad (\xi, t) \models \ \{x[t] < thr_{zone}\}, \\
(\xi, t) \models \text{“Unstable”} \quad &\Longleftrightarrow \quad (\xi, t) \models \ \mu_0 \wedge (\mu_1 \vee \mu_2 \vee \mu_3),
\end{aligned}
$$

where thr_{zone} is a CL threshold value of a given zone, programming recommendation for a thr_{zone} threshold is 40 ms. Predicates μ_0, μ_i $(i = 1, 2, 3)$ are defined as follows:

$$
\begin{aligned}
\mu_0 &:= \ \{\, VT_{counter}[t] > 3 \,\} \\
\mu_i &:= \ \{\, |x[t] - x[t-i]| > thr_{stab} \,\}
\end{aligned}
$$

All other building blocks will be defined in a same manner using all expressive power of TFL. One future research direction is a verification of the model $H \times D$ from the Figure 1 against the learnt TFL requirements using various automated techniques.

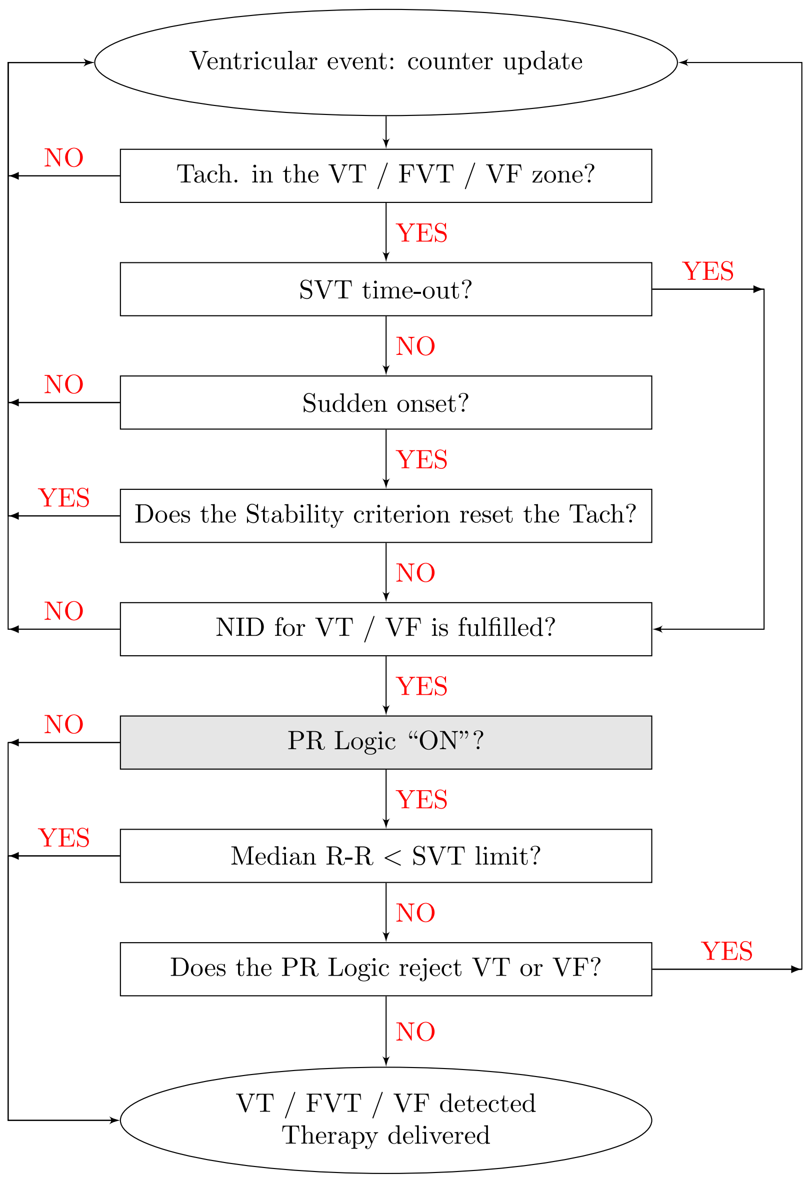

Figure 6: VT/SVT Discrimination: Medtronic dual chamber algorithm

5 Annual Report

I have commenced my doctoral studies on the 2^{nd} of February 2015 under the supervision of Radu Grosu and co-supervision of Agata Ciabattoni and Ezio Bartocci.

5.1 Coursework

	Title	ECTS	Type
Elective	Hybrid Systems	3	VU
	Model Predictive Control	3	VU
	Embedded Systems Engineering	3	VO
Compulsory	Introduction to Logical Methods in Computer Science	3	VO
	Research and Career Planning for Doctoral Students	3	SE
	Research Seminar LogiCS (ongoing)	3	SE

5.2 Current Progress

- Accepted paper: HSCC 2016 (19th International Conference on Hybrid Systems: Computation and Control): *Temporal Logic as Filtering* [18].
- Additionally, I am currently working on a journal paper, which is an extention of the work that was published in the HSCC 2016 [18]. Our plan is to submit it to the ACM Transactions on Cyber-Physical Systems (TCPS).
- Invited talk: *Temporal Logic as Filtering*
  ARVI Project Meeting. December 2015, Tallinn, Estonia
- Invited talk: *Listening to the Music of the Heart*
  CyberCardia Project Kick-Off Meeting. September 2015, Arlington, USA
- Talk: *On Temporal Logic and Signal Processing*
  CyberCardia Project WebEx Meeting. February 2015, Vienna, Austria
- External Journal Reviewing:
  - CMSB 2015 (Computational Methods in Systems Biology)
  - FORMATS 2015 (Formal Modelling and Analysis of Timed Systems)
  - TACAS 2016 (22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems)

- Schools participation: Automatic Verification and Analysis of Complex Systems 2nd AVACS Autumn School.
  September 30 - October 2, 2015, Oldenburg, Germany.

References

[1] Aristotle. *Physics*. The Internet Classics Archive by Daniel C. Stevenson, Web Atomics, 2009.

[2] G. H. Bardy, K. L. Lee, D. B. Mark, J. E. Poole, D. L. Packer, R. Boineau, M. Domanski, C. Troutman, J. Anderson, G. Johnson, S. E. McNulty, N. Clapp-Channing, L. D. Davidson-Ray, E. S. Fraulo, D. P. Fishbein, R. M. Luceri, and J. H. Ip. Amiodarone or an implantable cardioverterdefibrillator for congestive heart failure. *New England Journal of Medicine*, 352(3):225–237, 2005.

[3] E. Bartocci, F. Corradini, M. R. Di Berardini, E. Entcheva, S. A. Smolka, and R. Grosu. Modeling and simulation of cardiac tissue using hybrid i/o automata. *Theoretical Computer Science*, 410 (33–34):3149–3165, 2009.

[4] A. Donzé. Breach, a toolbox for verification and parameter synthesis of hybrid systems. In *Proc. of CAV 2010: the 22nd International Conference on Computer Aided Verification*, volume 6174 of *LNCS*, pages 167–170. Springer Berlin, 2010.

[5] A. Donze, O. Maler, E. Bartocci, D. Nickovic, R. Grosu, and S. Smolka. On temporal logic and signal processing. In *10th International Symposium, ATVA 2012, Thiruvananthapuram, India, October 3-6, 2012. Proceedings*, volume 7561 of *LNCS*, pages 92–106. Springer Berlin Heidelberg, 2012.

[6] A. Donzé, O. Maler, E. Bartocci, D. Nickovic, R. Grosu, and S. A. Smolka. On temporal logic and signal processing. In *Proc. of ATVA 2012: the 10th International Symposium on Automated Technology for Verification and Analysis*, volume 7561 of *LNCS*, pages 92–106. Springer, 2012.

[7] G. E. Fainekos, A. Girard, and G. J. Pappas. Temporal logic verification using simulation. In *4th International Conference, FORMATS 2006, Paris, France, September 25-27, 2006. Proceedings*, volume 4202 of *LNCS*, pages 171–186. Springer-Verlag Berlin, Heidelberg, 2006.

[8] R. Grosu, E. Bartocci, F. Corradini, E. Entcheva, S. A. Smolka, and A. Wasilewska. Learning and detecting emergent behavior in networks of cardiac myocytes. In *11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008. Proceedings*, volume 4981 of *LNCS*, pages 229–243, 2008.

[9] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2004.

[10] G. J. Klein, J. M. Gillberg, A. Tang, S. Inbar, A. Sharma, C. Unterberg-Buchwald, P. Dorian, H. Moore, F. Duru, E. Rooney, D. Becker, K. Schaaf, and D. Benditt. Improving svt discrimination in single chamber icds: A new electrogram morphology-based algorithm. *Journal of Cardiovascular Electrophysiology*, 17(12):1310–1319, 2006.

[11] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[12] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, 3253:152–166, 2004.

[13] O. Maler, D. Nickovic, and A. Pnueli. Checking temporal properties of discrete, timed and continuous behaviors. *Pillars of Computer Science*, 4800:475–505, 2008.

[14] Medtronic. Pacing defibrillation: Medtronic discrimination. url: http://pacingdefibrillation.com/en/, 2015.

[15] J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. In *6th international conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2008, Saint-Malo, France, September 1517, 2008. Proceedings*, volume 5215 of *LNCS*, pages 1–13. Springer-Verlag Berlin, Heidelberg, 2008.

[16] A. Pnueli. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57, 1977.

[17] M. Raitt. Implantable cardioverter-defibrillator shocks: A double-edged sword? *Journal of the American College of Cardiology*, 51(14):1366–1368, 2008.

[18] A. Rodionova, E. Bartocci, D. Nickovic, and R. Grosu. Temporal logic as filtering. In *19th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016, Proceedings*, 2016.

[19] S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama. Specmurt analysis of polyphonic music signals. *The IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 16(3):639–650, 2008.