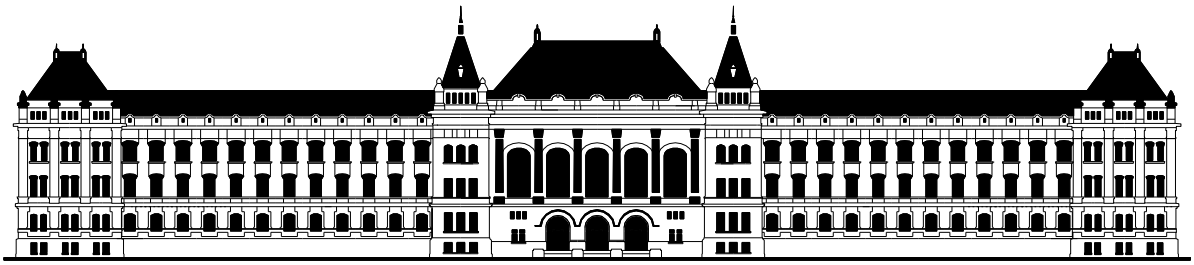
IEEE HUNGARY SECTION
CIRCUITS, SYSTEMS AND
COMPUTERS JOIN CHAPTER

# PROCEEDINGS
# OF THE
# 17TH PHD MINI-SYMPOSIUM

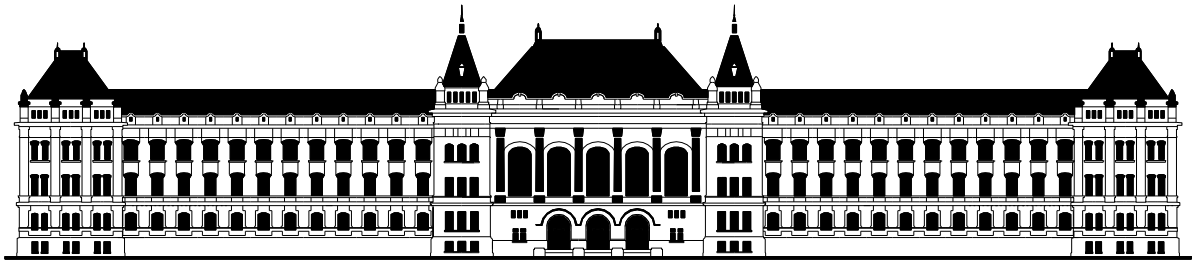
**FEBRUARY 1, 2010.**



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULITY OF ELECTRICAL ENGINEERING AND INFORMATION
DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

# PROCEEDINGS
# OF THE
# 17TH PHD MINI-SYMPOSIUM

**FEBRUARY 1, 2010.**
**BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS**
**BUILDING I**



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
FACULITY OF ELECTRICAL ENGINEERING AND INFORMATION
DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

Head of the Department: Prof. Dr. Gábor HORVÁTH

Conference Chairman:
Béla PATAKI

Organizers:
Sándor JUHÁSZ
Zsolt KOLLÁR
Imre PECHAN
István SZOMBATH
Zoltán UJHELYI

Homepage of the Conference:
http://www.mit.bme.hu/events/minisy2010/

Sponsored by:
IEEE Hungary Section
Circuits, Systems and Computers Joint Chapter

Schnell László Foundation

evosoft Hungary Kft.

Special supporter:


ISBN 978-963-420-994-2

# FOREWORD

This proceedings is a collection of the extended abstracts of the lectures of the 17th PhD Mini-Symposium held at the Department of Measurement and Information Systems of the Budapest University of Technology and Economics. The main purpose of these symposiums is to give an opportunity to the PhD students of our department to present a summary of their work done in the preceding year. It is an interesting additional benefit, that the students get some experience: how to organize such events. Beyond this actual goal, it turned out that the proceedings of our symposiums give an interesting overview of the research and PhD education carried out in our department. The lectures reflect partly the scientific fields and work of the students, but we think that an insight into the research and development activity of the department is also given by these contributions. Traditionally our activity was focused on measurement and instrumentation. The area has slowly changed during the last few years. New areas mainly connected to embedded information systems, new aspects e.g. dependability and security are now in our scope of interest as well. Both theoretical and practical aspects are dealt with.

The papers of this proceedings are sorted into four main groups. These are Embedded and Intelligent Systems; Information Mining and Knowledge Representation, Measurement and Signal Processing; Model-based Software Engineering,. The lectures are at different levels: some of them present the very first results of a research, because most of the first year PhD students have been working on their fields only for half a year, therefore they submitted two-page papers. The second and third year students are more experienced and have more results; therefore they have four-page papers in the proceedings.

During this seventeen-year period there have been shorter or longer cooperation between our department and some universities and research institutes. Some PhD research works gained a lot from these connections. In the last year the cooperation was especially fruitful with the Darmstadt University of Technology, Germany; IRISA Rennes, France; Ilmenau University of Technology, Germany; University of Firenze, Italy; IBM Budapest Center of Advanced Studies, Hungary; KFKI Research Institute for Particle and Nuclear Physics of the Hungarian Academy of Sciences, Budapest; Robert Bosch Kft., Hungary; National Instruments Hungary Software és Hardware Gyártó Kft., Debrecen; Innomed Zrt., Budapest.

We hope that similarly to the previous years, also this PhD Mini-Symposium will be useful for the lecturers, for the audience and for all who read the proceedings.

Budapest, January 13, 2010.

Béla Pataki

Chairman of the PhD Mini-Symposium

# LIST OF PARTICIPANTS

| Participant | Advisor | Starting Year of PhD Course |
|---|---|---|
| BÁNYAI, Mihály | STRAUSZ, György | 2009 |
| BERGMANN, Gábor | VARRÓ, Dániel | 2008 |
| DANCSI, György | DABÓCZI, Tamás | 2008 |
| ENGEDY, István | HORVÁTH, Gábor | 2009 |
| EREDICS, Péter | DOBROWIECKI, Tadeusz | 2009 |
| GUTA, Gábor | VARRÓ, Dániel | 2009 |
| HAJÓS, Gergely | ANTAL, Péter, DOBROWIECKI, Tadeusz | 2008 |
| HEGEDÜS, Ábel | VARRÓ, Dániel | 2009 |
| JUHÁSZ, Sándor | HORVÁTH, Gábor | 2007 |
| KOLLÁR, Zsolt | PÉCELI, Gábor | 2008 |
| KRÉBESZ, Tamás | KOLUMBÁN, Géza, DABÓCZI, Tamás | 2007 |
| LACZKÓ, Péter | FEHÉR, Béla | 2009 |
| OLÁH, János | VARRÓ, Dániel | 2009 |
| ORBÁN, Gergely | HORVÁTH, Gábor | 2009 |
| PALJAK, Gergely János | PATARICZA, András, KOVÁCSHÁZY, Tamás | 2009 |
| PECHAN, Imre | FEHÉR, Béla | 2008 |
| RAIKOVICH, Tamás | FEHÉR, Béla | 2007 |
| SÁRKÖZY, Péter | ANTAL, Péter, DOBROWIECKI, Tadeusz | 2009 |
| SZATMÁRI, Zoltán | MAJZIK, István | 2008 |
| SZOMBATH, István | PATARICZA, András | 2008 |
| UJHELYI, Zoltán | VARRÓ, Dániel | 2009 |
| VÖRÖS, András | BARTHA, Tamás | 2009 |

# Program of The MINI-SYMPOSIUM

# Conference Schedule

| | |
|---|---|
| 8:30 | Opening |
| 8:35 | Embedded and Intelligent Systems |
| 9:50 | Model-based Software Engineering |
| 12:10 | Lunch break |
| 13:30 | Measurement and Signal Processing |
| 16:00 | Information Mining and Knowledge Representation |

# UWB IMPULSE RADIO WITH GATED TRESHOLD COMPENSATED PPM RECEIVER

**Tamás KRÉBESZ**
**Advisors: Géza KOLUMBÁN, Tamás DABÓCZI**

## I. Introduction

General agreement has been reached recently, that the coherent optimum receivers known from the theory of conventional communication systems are not feasible to implement low-cost and robust Ultra-WideBand (UWB) Impulse Radio (IR) receivers, especially if they have to offer an extremely low power consumption. Instead, Energy Detector (ED)-based noncoherent receivers have to be used [1].

Unfortunately, the noise performance of the ED-based receivers is relatively poor compared to that of the coherent receivers, resulting in a low receiver sensitivity.

This contribution shows that a 7.7 dB improvement in receiver sensitivity can be achieved by disabling the receiver outside the UWB pulse duration. The low duty cycle offers an important extra advantage, namely, it reduces the receiver power consumption.

## II. The Noncoherent UWB IR PPM (Pulse Position Modulation) Transceiver

In the UWB impulse radio short impulses are used to carry the digital information. Due to their excellent spectral properties, frequency shifted Gaussian pulses are most frequently used as carriers

$$g(t) = \sqrt{\frac{Z_0 E_b}{\sqrt{\pi}\, u_B}} \exp\left(-\frac{t^2}{2u_B^2}\right) \cos(2\pi f_C t) \tag{1}$$

where $Z_0$ is the characteristic impedance over which the energy per bit $E_b$ is measured, $f_C$ denotes the center frequency of UWB pulse and the parameter $u_B$ is determined by the required 10 dB RF bandwidth which is $2f_B$ of UWB pulse

$$u_B = \frac{1}{2\pi f_B \sqrt{\log_{10}(e)}}. \tag{2}$$

In the latter equation $e$ denotes the base of natural logarithm.

### *A. Pulse Position Modulation*

Let $T_{bin}$ denote the time slot that is used to transmit one bit information. In PPM, the information is encoded into the position, $t_{pos1}$ or $t_{pos2}$, of the transmitted UWB pulse

$$s_m(t) = \begin{cases} g(t - t_{pos1}) & \text{for bit "1"} \\ g(t - t_{pos2}) & \text{for bit "0"} \end{cases}. \tag{3}$$

A great advantage is that the PPM signal can be demodulated by both coherent and noncoherent receiver.

As shown in Fig. 1, the bit duration $T_{bin}$ is divided into two identical time slots denoted by $T_{int1}$ and $T_{int2}$ in the UWB IR transceiver proposed in [2]. The position of the frequency shifted Gaussian pulse $g(t)$ is varied according to the bit to be transmitted, the duration of one UWB pulse is $T_{ch}$.

### *B. Threshold Compensated Noncoherent PPM Receiver*

The block diagram of threshold compensated noncoherent receiver [2] built by MIT (USA) using 90-nm CMOS technology is shown in Figure 2. To recover the transmitted bit, the receiver measures and compares the energy received in the two adjacent time slots denoted by $T_{int1}$ and $T_{int2}$ in Fig. 1.

The channel noise $n(t)$ corrupting the received signal $r_m(t) = g(t) + n(t)$ is suppressed by the channel filter $h(t)$ and the filter output $\tilde{r}_m$ is fed into a square-law device, then its output is integrated. The results of two integrations, that is, the energies received in the two adjacent time slots are stored in a sample-and-hold capacitor for bit slicing. The stored voltages are compared and the decision is done in favor of the larger received signal energy.

Compared to ED-based On-Off Keying (OOK) demodulator [1] the block diagram shown in Fig. 2 may seem to be a bit complicated. However, this configuration has a huge advantage, the optimum decision threshold is constant and does not depend on the SNR measured at the input of demodulator. Recall, in the ED-based OOK demodulators the optimum decision threshold depends on the SNR and it has to be varied adaptively according to the channel conditions.

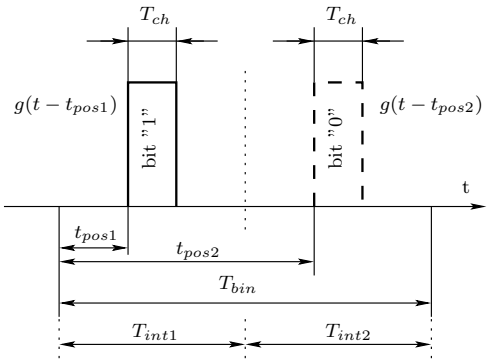



Figure 1: Modulated UWB IR PPM signal.

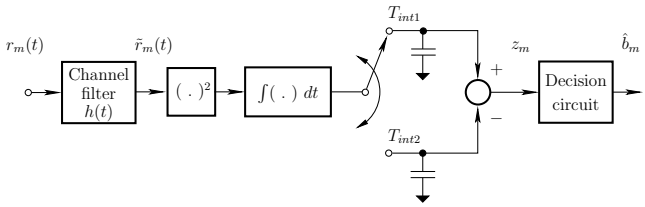



Figure 2: Block diagram of the threshold compensated noncoherent UWB IR PPM receiver.

## III. Noise Performance Improvement

### *A. Theoretical Noise Performance of ED-Type Demodulators*

Consider the threshold compensated PPM demodulator shown in Fig. 2. An analytical expression for the theoretical Bit Error Rate (BER) has been derived in [5] for the ED-type demodulators

$$P_e = \frac{1}{2^{2B\tau}} \exp\left(-\frac{E_b}{2N_0}\right) \times \sum_{i=0}^{2B\tau-1} \frac{\left(\frac{E_b}{2N_0}\right)^i}{i!} \sum_{j=i}^{2B\tau-1} \frac{1}{2^j} \begin{pmatrix} j+2B\tau-1 \\ j-i \end{pmatrix} \tag{4}$$

where the power spectral density of channel noise equals $N_0/2$, $2B$ and $\tau$ denotes the receiver bandwidth and the energy capture time, respectively. Although (4) had been developed for the Transmitted Reference (TR) transceiver in chaotic communications, later it was shown that (4) is valid for any kind of carriers $g(t)$ including UWB pulses [6], conventional sinusoidal carriers, and chaotic carriers [5] provided that $E_b$ is kept constant.

It was confirmed in [1] that the BER of a TR system and a PPM system with energy detector are identical. Consequently, (4) is valid to describe the noise performance of the noncoherent UWB IR PPM transceiver.

I determined the noise performance of threshold compensated noncoherent UWB IR PPM receiver as plotted in Fig. 3 where $2B\tau$ is chosen as parameter. $2B\tau$ is set to 2 (solid curve), 25 (dashed curve) and 250 (dotted curve). Curves show the theoretical BER calculated from (4), while marks '+' give the results of simulations. As expected, the product of $2B\tau$ has a very serious influence on the noise performance, to get the best receiver sensitivity $2B$ and $\tau$ have to be matched to the bandwidth, $\sim 2f_B$, and duration, $\sim T_{ch}$, of the UWB pulse $g(t)$, respectively.

### *B. Gated Threshold Compensated UWB IR PPM Receiver*

The block diagram of the gated threshold compensated noncoherent UWB IR PPM receiver is shown in Fig. 4 where the channel filter matches the receiver noise bandwidth to the bandwidth of UWB carrier pulse and the two gates disable the receiver outside the duration of UWB carrier pulse.

## IV. Optimal Fitting of the Receiver Parameters

To illustrate the efficiency of performance improvement technique proposed here let us consider an IEEE Std 802.15.4a-compliant UWB IR PPM system with UWB bandwidth of 499.2 MHz and data rate of 1 Mbit/s. Assuming that one waveform is transmitted for one bit then $T_{bin} = 1000$ ns.

The frequency-shifted Gaussian UWB pulse is limited neither in the time- nor in the frequency-domains. However, the receiver has a fixed noise bandwidth and observes the received signal for a finite time period, consequently, a slight loss in the reception of UWB signal energy per bit is inevitable. The optimization of receiver parameters is performed in two steps: (i) during the coarse fitting, $2B$ and $\tau$ are optimized using (4). Since the analytical expression is valid only for integer values of $2B\tau$ it makes possible only a coarse fitting of receiver parameters. (ii) During fine fitting, a computer simulation is used to find the optimal values of $2B$ and $\tau$.

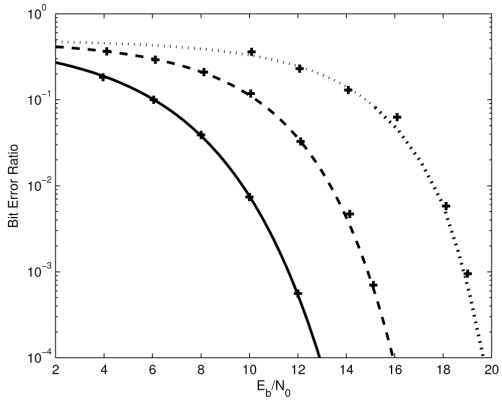



Figure 3: Noise performance of threshold compensated noncoherent UWB IR PPM receiver when $2B\tau$ is set to 2 (solid curve), 25 (dashed curve) and 250 (dotted curve). Curves show the theoretical results calculated from (4), while marks '+' give the result of simulations.

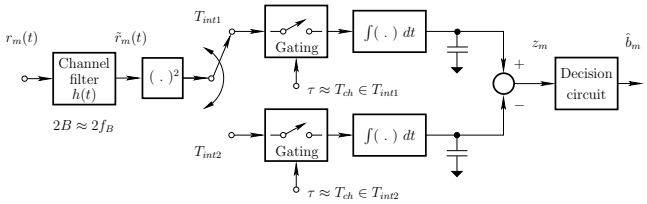



Figure 4: Block diagram of the gated threshold compensated noncoherent UWB IR PPM receiver.

### *A. Coarse Fitting of Receiver Parameters*

The receiver noise bandwidth has to be wide enough to pass the received UWB signal without a considerable loss in $E_b$. Since the bandwidth of IEEE Std 802.15.4a-compliant UWB signal is 499.2 MHz, let $2B = 500$ MHz be chosen. If the data rate is 1 Mbit/s and the *energy capture time is not fitted* then $\tau = T_{bin}/2 = 0.5$ $\mu$s and $2B\tau = 250$. As shown by the dotted curve in Fig. 3 if a bit error ratio of $10^{-3}$ has to be achieved by these receiver parameters then $E_b/N_0 = 19$ dB has to be assured at the input of the proposed PPM demodulator.

Due to the short duration of transmitted UWB pulses, the energy capture time may be reduced considerably without loosing a noticeable part of $E_b$. Our investigations have shown that the energy capture time can be reduced to $4$ ns.

Let the receiver noise bandwidth kept unchanged, that is, $2B = 500$ MHz, but let the *energy capture time be reduced* to $\tau = 4$ ns. Then $2B\tau$ becomes 2 and, as shown by the solid curve in Fig. 3, the required $E_b/N_0$ becomes 11.6 dB. Note, a 7.4 dB improvement has been achieved in the demodulator noise performance and in the receiver sensitivity by fitting the energy capture time to the duration of UWB carrier pulse.

### *B. Fine Fitting of Receiver Parameters*

The frequency shifted Gaussian pulse is decaying smoothly both in the frequency and time domains. Equation (4) is valid only for integer values of $2B\tau$, it cannot take into account the smooth decay of UWB pulse. For example, if the receiver bandwidth $2B$ is further reduced then a part of $E_b$ is lost but, simultaneously, a part of channel noise is also suppressed. The optimum value of $2B$ is a trade-off between the two effects. An extra improvement in noise performance can be achieved if the optimum values of $2B$ and $\tau$ are determined by computer optimization.

A raw BER of $10^{-3}$ has to be achieved in the majority of WPAN applications. According to Fig. 3, this BER requires an $E_b/N_0 \approx 12$ dB at the input of the noncoherent gated threshold compensated PPM demodulator. To check the effect of fine tuning of $\tau$ on the performance, $E_b/N_0$ is set to 12 dB.

Figure 5 shows the effect of energy capture time on the BER where the receiver noise bandwidth was set to 500 MHz. Observe, the energy capture time has to be reduced to 3 ns to get the best receiver noise performance.

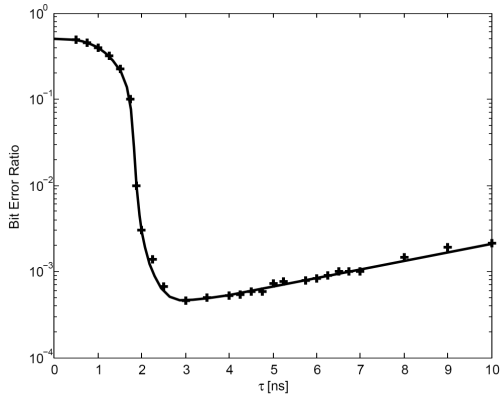



Figure 5: Effect of energy capture time on the noise performance for $2B = 500$ MHz and $E_b/N_0 = 12$ dB. Results of simulation are marked by '+'.

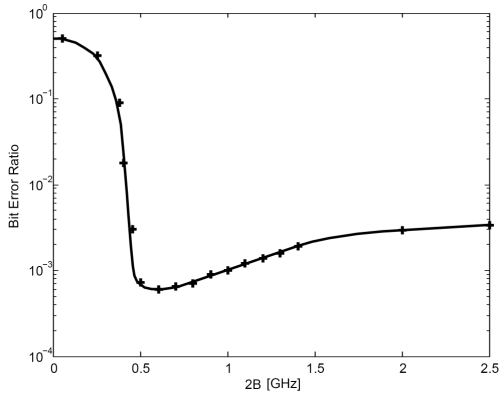



Figure 6: Effect of receiver noise bandwidth on the noise performance for $\tau = 4$ ns and $E_b/N_0 = 12$ dB. Results of simulation are marked by '+'.

The effect of receiver noise bandwidth on the BER are plotted in Fig. 6 where the energy capture time was set to 4 ns. Note, to get the best noise performance the receiver noise bandwidth has to be slightly increased, its optimum value is about 600 MHz.

Applying the fine fitting of the receiver parameters further 0.3 dB noise performance improvement can be achieved. Summing up the result of both coarse and fine fitting we conclude that a 7.7 dB improvement in receiver sensitivity has been achieved.

## V. Conclusion

The low-rate UWB impulse radio operates with an extremely low duty cycle. The paper has shown how this low duty cycle can be exploited to improve the receiver noise performance and its sensitivity. By fitting the energy capture time and noise bandwidth of the noncoherent demodulator to the parameters of UWB carrier pulse, a 7.7 dB improvement has been achieved in the receiver sensitivity. That improvement enables the application of cheap CMOS gated threshold compensated UWB IR receiver in many new LR-WPAN applications.

## References


[1] K. Witrisal, G. Leus, G. J. M. Janssen, M. Pausini, F. Troesch, T. Zasowski, and J. Romme, "Noncoherent Ultra-Wideband Systems: An Overview of Recent Research Activities," *IEEE Signal Processing Magazine*, 26(4):48–66, July 2009.

[2] P. P. Mercier, D. C. Daly, M. Bhardwaj, D. D. Wentzloff, F. S. Lee, and A. P. Chandrakasan, "Ultra-low-power UWB for sensor network applications," in *ISCAS'08*, pp. 2562–2565, Seattle, Washington, USA, May 18-21 2008.

[3] Federal Communications Commission, *Part 15 of the Commission Rs Rules Regarding Ultra-Wideband Transmission Systems; Subpart F*, FCC–USA, Online: <http://sujan.hallikainen.org/FCC/FccRules/2009/15/>.

[4] IEEE Std 802.15.4a-2007, 2007.

[5] G. Kolumbán, "Theoretical noise performance of correlator-based chaotic communications schemes," *IEEE Trans. Circuits and Systems—Part I: Fundamental Theory and Application*, 47(12):1692–1701, December 2000.

[6] G. Kolumbán and T. Krébesz, "UWB radio: A real chance for application of chaotic communications," in *Proc. NOLTA'06*, pp. 475–478, Bologna, Italy, September 11–14 2006.

# RECEIVER SYNCHRONIZATION FOR 2-FSK COMMUNICATION SYSTEMS

**Zsolt KOLLÁR**
**Advisor: Gábor PÉCELI**

## I. Introduction

In this paper a robust and effective synchronization technique is presented for a low transmit data rate communication using frequency shift keying (2-FSK). Due to security purposes the designed system has to be very stable with low packet error rate and should be capable to operate at low signal to noise ratios (SNR).

The main idea of digital communication systems is to transmit binary information through the medium. A schematic block diagram of a digital communication system is shown in Fig.1. First the the bitstream is mapped to baseband analog signal. Then the carrier signal is modulated by the baseband signal and transmitted after a high power amplifier (HPA) trough the medium. Depending on which parameter of the carrier wave is manipulated, we can talk about amplitude, phase or frequency modulation [1]. At the receiver we use first apply a low noise amplifier, the other steps are the inverse of the above described steps. The baseband signal processing blocks (shaded blocks) will be the main topic in this paper. Although it is not shown in the figure, a key issue is the synchronization of

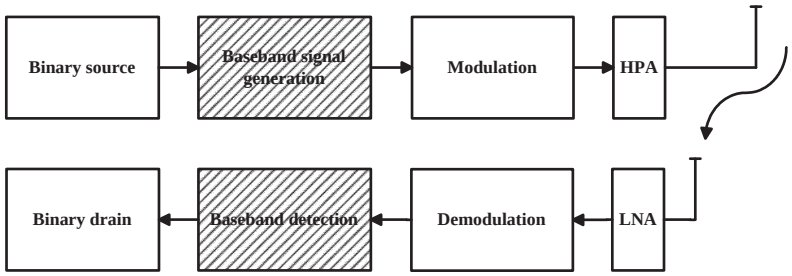



Figure 1: Block diagram of simplified digital communication system

the received signal at the receiver. Timing, phase and frequency synchronization must be performed to retrieve the correct data [2]. In this paper we will show a synchronization method for a simple system using frequency shift keying (FSK) in the baseband. Modulation in our case means a simple up-converting to the desired transmit channel.

As baseband signalling we use frequency shift keying (FSK). A 2-FSK system transmits one bit at time over a symbol length $T_s$, using 2 frequencies $f_1$ and $f_2$ with the following elementary analog symbols:

$$s_1(t) = \sin(2\pi f_1 t), \quad 0 < t < T_s, \quad \text{representing bit "0"}$$
$$s_2(t) = \sin(2\pi f_2 t), \quad 0 < t < T_s, \quad \text{representing bit "1"}$$

The frequency shift keying modulation is one of the most robust modulation techniques, therefore this is one of the best choices for transmitting in a noisy channel. In our case the baseband frequency bandwidth is 3600 Hz.

## II. Structure of the baseband transmission signal

To comply with the international standards we chose the following 2 frequency values for the elementary symbols: 1200 Hz and 2400 Hz. It is also important to mention that a data rate of 2400 bps had to be achieved, so a total period of the 2400 Hz signal is applied and a half period of the 1200 Hz signal. To keep the signal continuity between consecutive elementary symbols, we also use the inverted versions of the elementary symbols for transmission. A transmission signal is constructed from these major parts:

- Due to the properties of the receiver, for a certain time a signal has to be transmitted to open the squelch of the receiver radio (Squelch).
- A synchronization pattern has to be transmitted so the symbol borders can be extrapolated (Synchronization bytes). This pattern is a continuous alternation between the two elementary symbols.
- A given bit pattern has to signal the beginning of the packet (Flag). The flag pattern is 01111110.
- The packet contains the data bits (Packet). It is important that it does not contain any flag pattern, this can be eliminated by bitstuffing (after every fifth 1 bit a 0 is inserted).
- A given bit pattern has to signal also the end of the packet (Flag).

An example for a transmission signal – with all the above described properties – is visualized in Fig. 2. Also the inverted elementary symbols for keeping the time domain continuity can be observed.

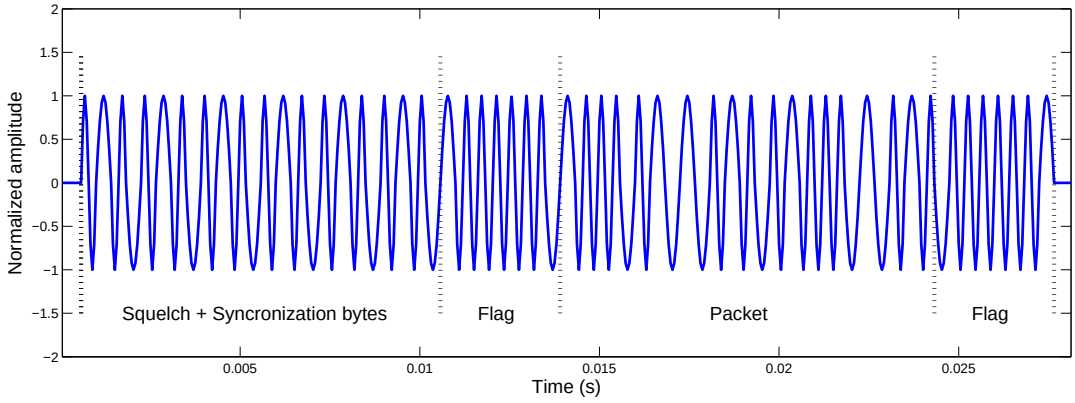



Figure 2: Structure of the baseband transmission signal

## III. The synchronization and demodulation algorithm

### A. *Correlation principle*

The receiver has to perform the synchronization based on the samples taken from the received signal. The synchronization algorithm is based on the correlation principle [3]. The discrete cross correlation between the incoming signal and the ideal elementary symbols has to be calculated. The two elementary symbols are orthogonal to each other for the symbol length:

$$\int_0^{T_s} s_1(t)s_2(t) = 0.$$

In the discrete case, where we have N samples from a symbol, we obtain: $\sum_0^{N-1} s_1[k]s_2[k]$. The receiver algorithm will take advantage of this property to decide between the two symbols.

*B. The synchronization algorithm*

The block diagram of the receiver structure with the synchronization algorithm is depicted in Fig. 3. First, a sample is taken from the incoming baseband analog signal. In general $N = 2^m$ samples are

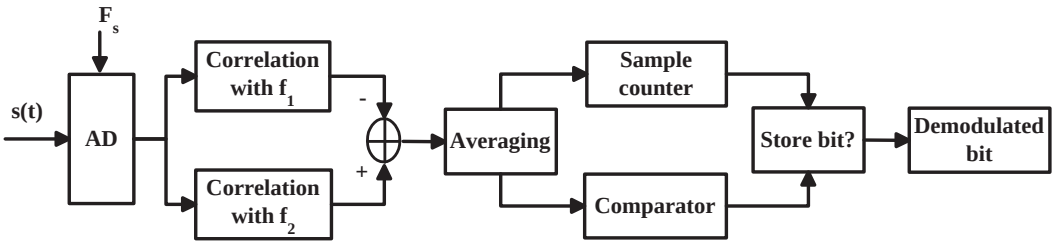



Figure 3: Block diagram of the receiver and synchronization algorithm

taken from one symbol length $T_s$, so the sampling frequency is equal to $F_s = N * 2400$ Hz. Then a filtering – in other words, a cross correlation calculation with $s_1$ and $s_2$ and with their cosine versions – is preformed between the incoming samples and the ideal ones. In the next step the difference of the two correlation results is calculated. With some restrictions it can be also interpreted as an N point Fourier transform [4] for two frequency values, and the difference between the two amplitude values is formed. This difference is then averaged to suppress the effect of noise.

The sample counter can have the values of 0 . . . N-1. If the sample counter reaches $N - 1$, a decision can be performed. If the sign of the output of the averaging filter block is changing, we can assume that we are at the half of a symbol length, so we can set our sample counter to $N/2$. On the other hand, if the absolute value of the output is smaller then a threshold – this threshold for the noise was determined by simulations –, then we can assume that we did not receive any signal, only noise, so we don't have to make any decisions. The sign of the averaging filter determines the demodulated bit, if it is negative, than we have a 0 bit, otherwise a 1 bit is received. After each decision the sample counter value is set back to 0.

Due to averaging, we will have a delay of one symbol length between the decision and the current incoming symbol. Synchronization of the symbol borders are done in every case when there is a change in the bits, due to the fact that averaging changes its sign, so the sample counter is set to the correct value.

If only 0 bits are transmitted, it will demodulate but in case of sampling time differences it can lead to loss of bits, due to the fact that no changes are inside the packet, so no sample counter correction is performed. This problem can be eliminated by introducing an extra NRZ (Non Return to Zero) coding to the bitstream inside the packet. Another drawback of this synchronization principle is that it is sensitive to small amplitude offsets, because the correlation with the $s_0$ symbol will provide a small value, but the correlation with $s_1$ totaly eliminates it. To improve the algorithm, the number of samples N taken from the incoming signal can be increased, although it is important to execute the synchronization algorithm between two sampling instants.

The output of the averaging filter for a signal to noise ratio of 10 dB is shown in Fig. 4. It can be seen that if only noise is present it has no influence on the detection, although decisions are made, but due to the fact that the values are under the decision threshold these are neglected. The two flags – packet start and packet end – can be also recognized. The receiver recovers the transmitted data for the received signal without errors. The normalized output of the averaging filter is set so, that without noise it reaches only 75% of the maximum value (maximum value is show with dashed lines at 1 and -1), this will lead to a margin for the noise effects. The threshold for the noise is shown by 2 doted lines at the value 0.125.

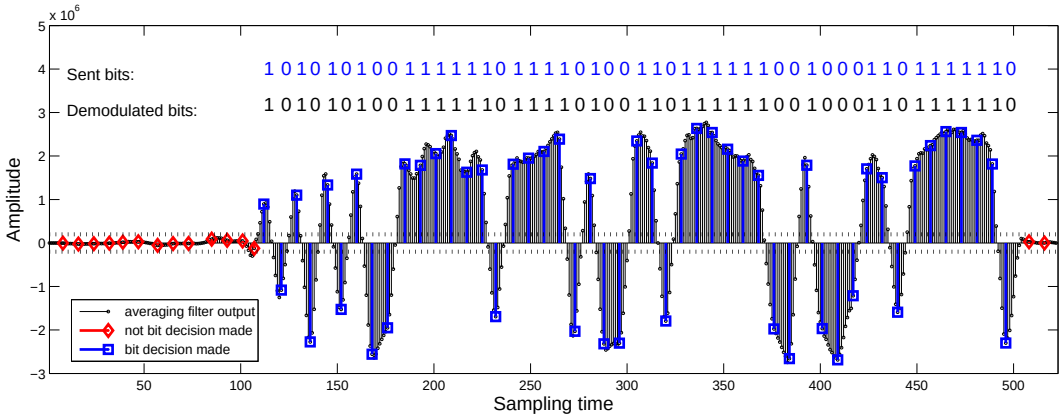



Figure 4: Output of the averaging block with decisions

### *C. Demodulation algorithm*

If the algorithm achieves synchronization it begins to demodulate the received bits. The incoming bits are stored until the packet start pattern is found: $01111110_2$. Then, the algorithm stores the incoming bits until the closing pattern is found. After the closing flag arrives, the packet is processed.

## IV. Conclusion

We have described robust synchronization algorithm for 2-FSK system. We have approved with simulation that it is effective if the SNR ratio is higher then 5 dB, and we have also shown that it is not severely affected by small timing offsets. The above described algorithm is also implemented and tested in DSP.

## Acknowledgements

I would like the express my thanks to all the members of the Rohde & Schwarz Reference Laboratory for their support and help on this topic. I'm especially grateful to Csaba Szombathy for his helpful observations regarding this paper.

## References

[1] J. Proakis, Ed., *Digital Communciations*, McGraw-Hill, 2000.

[2] Rohde and Whitaker, Eds., *Communications Receivers*, McGraw-Hill, 2001.

[3] J. Min and et al., "Low power correlation detector for binary fsk direct-conversion receivers," *Electronics Letters*, 31:1030–1032, June 1995.

[4] S. Hara and et al., "A novel fsk demodulation method using short-time dft analysis for leo satellite communication systems," *IEEE Trans. Veh. Technol.*, 46:625–633, Aug. 1997.

# PREPROCESSING FOR LUNG CONTOUR DETECTION

**Sándor JUHÁSZ**
**Advisor: Gábor HORVÁTH**

## I. Introduction

Chest radiographs can help the detection of lung cancers, TB and other lung diseases. The early detection of cancers is very important as it significantly increases the chance to cure the disease.

Global screening is a relatively cheap way to detect breast cancers and TB. Such screening programs generate a vast amount of pictures to be analyzed by experts. This is where computer-aided detection (CAD) can help the work of radiologists.

CAD systems are not reliable enough to do the work alone at the moment. The current goal is only to create a system that can help the detection and increase the accuracy of examination by searching suspicious areas (region of interest, ROI) or by enhancing the visibility of the picture at the darker regions. The first step in such an evaluation would be the delineation of the organs' boundaries. Searching the contour of the lung, the heart, the clavicles and the ribs determine the area of processing and the raw data for image enhancing. The lung contour has diagnostic value without further processing too as it can show cardiomegaly and pneumothorax.

## II. Contour Detection

Several methods have been developed to search the contour of the lungs. The problem is difficult for computers as the pictures are sometimes noisy and different parts of the body overlap on the x-ray images. Edges of the ribs and the clavicles and the boundaries of the breasts make the contours less clear and add further edges making proper contour detection more difficult.

Some algorithms try to get the lung area by classifying each pixel. Information of the surrounding texture, the position of the pixel and the classification of neighbouring pixels can be used. These methods usually use *neural networks*, *support vector machines* or *kNN-classifiers*. After getting a rough result from these algorithms further processing is usually needed to get lung-like results by making the area connected and avoiding holes.

Another group of algorithms concentrate on the contour only. These are usually *snake*-like methods [4]. Active Shape Model [5] seems the most successful approach for lung contour detection in this algorithm family. It starts from an average contour and moves contourpoints to fit the actual shape in each iteration. If the initial contour is too far from the real contour then the algorithm will fail to converge to the correct solution. To solve this problem the training contours are normalized with Procrustes-analysis [7] and some preprocessing is used to determine the position and orientation of lung. After that the ASM has to run only in the selected region.

Some authors [1] state that ASM gives better results without using Procrustes-methods. This can be true for databases where the position, size and orientation of the lungs show little variance. Unfortunately this is not always the case in general practice. Figure 1. shows a few example images where we can see that the lungs are sometimes very different in size and position. Sometimes a small part of the lung is not even on the image.

In these difficult cases algorithms that start from an average contour give false results often. That's why some preprocessing is needed to determine a better initial position.

Figure 1.a

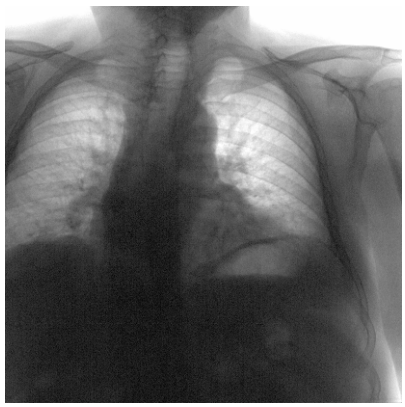


Figure 1.b

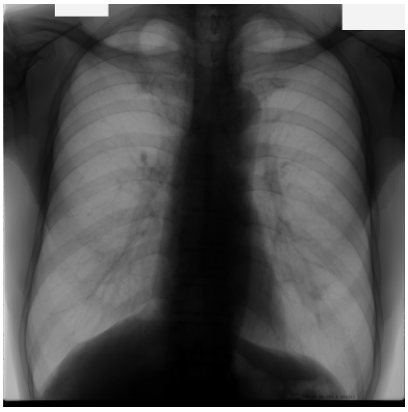


Figure 1.c

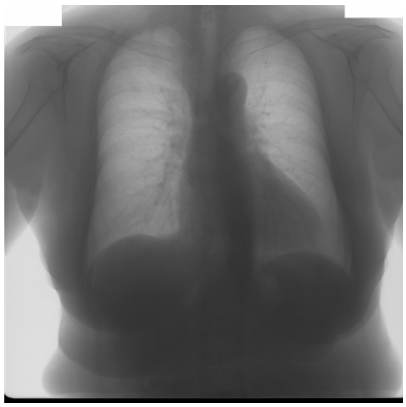


Figure 1.d

Figure 1: Lungs with different sizes, shapes, positions and angles on x-ray images

## III. Boundary Box Detection

As the detection of the lungs' boundary box helps the contour detection algorithms, it is useful to create a method that can approximately predict the position of the boundary box. Several algorithms are known that can recognize objects with high shape and texture variety. We used a method very similar to [2].

### *A Training*

To train the algorithm we need images where we already know the boundary box of the lungs'. We don't need the exact lung contours as our goal will be only to find a good starting position for the more sophisticated algorithms. We divide the region in the boundary box (or a slightly increased region to avoid false positive answers inside the good area) into *Tw* horizontal and *Th* vertical parts to get *Tw*Th* tiles. On Figure 2 we can see an example division. From each tile we extract F texture features.

We collect all these features and create one descriptor vector that describes the region of the boundary box. It will have *Tw*Th*F* dimensions. This way we can get one positive example per image. Creating negative examples is much easier. We can sample the images at other positions and we can use different boundary box sizes, so we will have plenty of negative vectors per image.

We train a two-class SVM classifier [6] to separate the positive and negative vectors. We can balance the number of positive and negative vectors by using the positive ones multiple times in the training set.

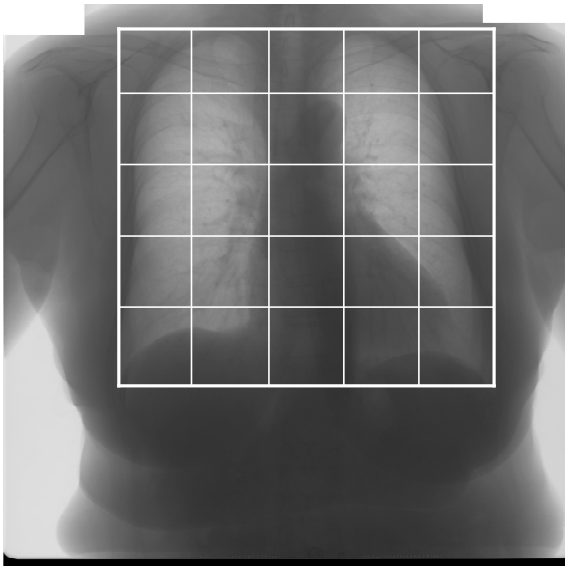


Figure 2: An example of division of the lungs' boundary box into 5*5 tiles

*B Detection*

During the detection we use the trained SVM classifier. We use a sliding-window technique. We slide a window with the average boundary box size over the image, calculate the feature vector at each position and evaluate the vector with the SVM classifier. We repeat it with different box sizes (both width and height) and choose the best fit.

*C Possible Features*

We should choose features that can represent the structure of the lung and have different values inside the lung and in other tissues outside the lung area.

One possible choice for the features comes from edge detection. We can apply an edge detector with a threshold (for example Canny) at various scales to the normalized image. We use the number of connected edges or the sum of edge lengths at each scale as a feature. It is useful because the lung area is brighter than the surrounding body parts, inside the lung many small structures can be seen (bronchial tree, veins, ribs) while outside the lung small structures are less visible. An edge detector for the size of these structures will give more results inside the lung than outside and thus gives a chance to separate these regions. At larger scales the strong edge of the lung contour indicates the boundary of the region of interest.

We can normalize the edge lengths by the area or by the side length of the examined tile. The first is more useful at smaller scales where we would like to measure the general small structure density. The latter works better at higher scales where the strong edge of the dark and bright region boundary is detected.

## IV. Integral Histogram Speed-up

At the detection phase we have to calculate the edges at different scales only once and use the results to compute the feature vectors for each test window. The latter calculation can be very time-consuming as we must compute the vectors at each position and scale of the sliding window. A speed-up technique called integral histogram for such problems has been developed (see [3] for details).

Instead of counting the sum of feature values inside a tile each time we need to create the feature vector, we can use a few easy operations after some preprocessing. Let $f(x,y)$ be the value of a feature at a certain position of the image. We define the integral of $f(x,y)$ as

$$I(x,y) = \sum_{i=0}^{x}\sum_{j=0}^{y} f(i,j)\,. \qquad (1)$$

When we calculate a feature's sum inside the tile we can simply use the precalculated *I(x,y)* values at the corners of the tile. If the top left corner of the region is $(x_1, y_1)$ and the bottom right corner is $(x_2, y_2)$ then the following equation gives the result:

$$F(x_1, y_1, x_2, y_2) = I(x_2, y_2) + I(x_1 - 1, y_1 - 1) - I(x_1 - 1, y_2) - I(x_2, y_1 - 1)\,. \qquad (2)$$

It's only an addition and two subtractions instead of a sum over the whole area. Figure 3 shows the areas involved in the calculation.

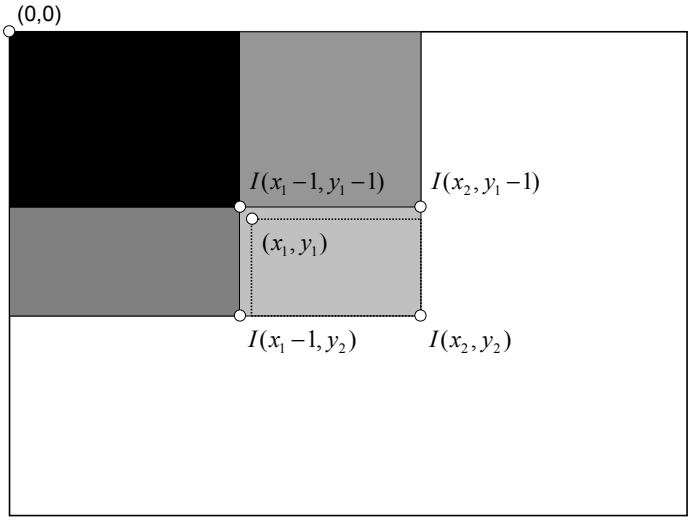



Figure 3: the integrals used to evaluate a tile

## V. Conclusion

In this paper we presented a technique that can determine the approximate boundary box of the lungs. This information is a big help for ASM algorithms when the position or the size of the lungs are far from the average.

In the future we plan to determine the optimal parameters of the algorithm and optimize it for GPUs.

## References


[1] B. van Ginneken, M. B. Stegmann and M. Loog, "Segmentation of Anatomical Structures in Chest Radiographs using Supervised Methods: A Comparative Study on a Public Database," *Medical Image Analysis*, 10(1):19-40, 2006.

[2] V. Ferrari, L. Fevrier, F. Jurie and C. Schmid, "Groups of Adjacent Contour Segments for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):36-51, Jan. 2008.

[3] F. Porikli, "Integral Histogram: A Fast Way To Extract Histograms in Cartesian Spaces," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:829-836, June 2005.

[4] M. Kass, A. Witkin, D. Terzopoulos, *"*Snakes: Active contour models,*" International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-331, 1988.

[5] T.F. Cootes, C.J. Taylor, D. Cooper, J. Graham, *"*Active shape models – their training and application,*" Computer Vision and Image Understanding,* 61 (1), 38-59., 1995.

[6] V. Vapnik, *The Nature of Statistical Learning* Theory, Springer, 1998.

[7] C. Goodall, *"*Procrustes methods in the statistical analysis of shapes,*" Journal of the Royal Statistical Society B*, vol. 53, no. 2, pp. 285-339, 1991.

# DETECTION OF LUNG NODULES ON CHEST RADIOGRAPHS

**Gergely ORBÁN**
**Advisor: Gábor HORVÁTH**

## I. Introduction

Lung cancer is one of the most common causes of cancer death. Many cures are known, but most of them are effective only in the early and symptomless stage of the disease. Screening can help early diagnosis, but an accurate, cheap and side effect free method has to be used to enable mass usage. Standard chest radiography mostly meets these requirements, except that current methods have a moderate accuracy. If someone suffering from cancer undergo the screening procedure, only has an approximately 30% probability of being diagnosed as positive. Efficiency can be improved by analysing the radiographs using a CAD (Computer Aided Detection) system. The most important problem of existing CAD systems is the high number of false detections. Although they can detect 60-70% of cancerous tumours, they also mark approximately four healthy regions on each image [1].

The goal of my work is to create a lung CAD being more effective than existing ones. For my current solution I used a common two step scheme utilizing my own improvements in each step. The first step is the enhancement of the target lung nodules by using various image processing algorithms. The second step selects suspicious areas on the enhanced image with the help of a classifier.

## II. Constrained Sliding Band Filter

The aim of the first step in the scheme is the enhancement of nodules on chest radiographs. These nodules are darker than the surroundings and round shaped. A commonly used filter family called CI (Convergence Index) enhance areas based on their shape. A common property of round shaped objects is the radial direction of gradient vectors along their border, so the filters consider the surrounding of the centre. The output depends on the angle of the gradient vectors and the vectors connecting the centre and the given points. One of the most successful realization is the SBF (Sliding Band Filter) using the following idea [2] and illustrated on figure1. For a given centre it slides a band in each direction within given bounds, while the band has a fixed width. For each band the algorithm sums the cosine of the angles of radial and gradient vectors. The final position of the band for a direction will be the one with the highest sum. Finally it sums the maximal band values in each direction. A high final sum indicates a nodule. A weakness of the algorithm is the independence of the bands in each direction, enhancing very spiculated and distorted objects. My proposed algorithm the CSBF (Constrained Sliding Band Filter) links the position of the bands allowing smaller distortion. It ensures that the final band positions satisfy a circularity constraint controlled by a coefficient. The enhanced pixel values can be calculated with the following formula.

$$CSBF(x,y) = \max_{R_{min} \le r \le \frac{R_{max}}{c}} \frac{1}{N} \sum_{i=1}^{N} Cmax_{ir}, \quad (1)$$

$$Cmax_{ir} = \max_{r \le n \le r*c-d} \frac{1}{d} \sum_{m=n}^{n+d} \cos\theta_{im}, \quad (2)$$

where $R_{min}, R_{max}$ are the bounds of the target object radius, $c$ is the shape constraint coefficient, $N$ is the number of directions concerned, $d$ is the width of the band and $\theta_{im}$ is the angle of the radial unit

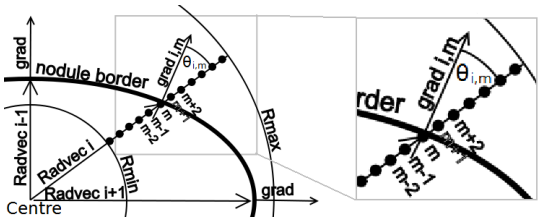



Figure 1: The CSBF filter.

vector and the m[th] gradient vector along the i[th] radial direction. The same result can be achieved by running several SBF filterings with different bounds and taking the minimum for each centre, however the execution times would be much greater, while CSBF does not take longer than one SBF run. The CSBF is a more general algorithm, thus for certain $c$ values it works as a standard SBF. Furthermore for $c = 1$ the CSBF is identical to the Iris filter, another realization of the CI family.

## III. False Positive Reduction

The second step of the CAD scheme concerns the areas with high CSBF value. A good practice is to collect many areas and select the suspicious ones with the help of a classifier, here an SVM (Support Vector Machine)[3]. The training sample set is extracted from a radiograph database with validated nodules. The raw input of the classifier is a 140 dimension vector containing various features that describe the shape, texture and symmetry of the area. Before handing the vector to the SVM, it is reduced to approximately 10 uncorrelated and relevant dimensions. For the SVM itself, I use a radial kernel function and my own parameter selection techniques.

## IV. Results

For testing I used a database of 247 images widely used for benchmarking created by the JSRT (Japanease Society of Radiological Technology) and a private database of 150 images originating from a Hungarian clinic. As a test method I used 4-fold cross-validation. On the JSRT database 59% of the real nodules were found while producing on average 2.5 false detections per image which equals to the performance of the most efficient method in my scope and using the JSRT database [1]. The CSBF contributed to the performance by an approximately 5% increase in sensitivity and same specificity compared to my previous solution using SBF. On the other hand the lack of accurate nodule segmentation in my system is the probable reason why it cannot outperform the existing best solution of others. The performance on the private database are somewhat worse. The sensitivity is 60% at a false positive rate of 4. The main reason behind the results is the greater variety of the private database.

## V. Conclusion

In conclusion the proposed system is capable of helping lung cancer diagnosis. To prove the usability, the system is built in the software of an X-ray machine and used experimentally at a clinic. The performance has to be improved further to finally provide the radiologists a really efficient tool.

## References

[1] R. C. Hardie, S. K. Rogers, T. Wilson, and A. Rogers, "Performance analysis of a new computer aided detection system for identifying lung nodules on chest radiographs," *Medical Image Analysis*, 12(3):240–258, June 2008.

[2] C. S. Pereira et al., "Evaluation of contrast enhancement filters for lung nodule detection," in *ICIAR 2007 Proceedings*, pp. 878–888, Montreal, Canada, Aug. 22–24 2007.

[3] M. Altrichter, G. Horváth, B. Pataki, G. Strausz, G. Takács, and J. Valyon, Eds., *Neurális Hálózatok*, Panem, 2006.

# CONTEXTUAL GRAPH TRIGGERS

**Gábor BERGMANN**
**Advisor: Dániel VARRÓ**

## I. Introduction

Model transformations play a crucial role in modern model-driven system engineering. Tool integration based on model transformations is a challenging task of high practical relevance, as it aims to harmonize the entire toolchain into a streamlined workflow. In tool integration scenarios, a complex relationship needs to be established and maintained between models conforming to different domains and tools. This *model synchronization* problem can be formulated as to keep a model of a *source language* and a model of a *target language* consistently synchronized while developers constantly change the underlying source and target models. Model synchronization is frequently captured by *transformation rules*.

Graph transformation (GT) [1] is a mathematical formalism frequently used for model transformation and other purposes; models are represented as (typed, attributed) graphs, and model manipulation is specified by *graph transformation rules*. GT rules consist of two *graph patterns*, and describe a change in the graph where an occurrence of the first pattern (precondition or left-hand-side, LHS) is replaced with the occurrence of the other pattern (postcondition or right-hand-side, RHS).

Traditionally, model transformation tools support the *batch execution* of transformation rules, which means that the input model is processed "as a whole", and output is either generated again from scratch, or, in more advanced approaches, updated using trace information from previous runs. However, models are *evolving* and changing continuously. In case of large and complex models used in agile development, batch transformations may not be feasible. To address this problem, live transformation [2] is an execution mode where changes to source models can be instantly mapped to changes in target models. Live transformations are persistent and go through event-driven phases of execution whenever a model change occurs.

Live transformation is based on event-driven rules. Some approaches including [2] regard only elementary model changes as events. Using a GT-based approach as a more general formalism of live transformations, [3] presented the novel formalism of *graph triggers* to capture high-level change events. Graph triggers are annotated GT rules that are executed upon the appearance or disappearance of pattern matches. These macroscopic events drive the execution of the transformation, independently of the low-granularity individual change that completed it. While monitoring the applicability of graph triggers is an involving task, it can be implemented efficiently with *incremental pattern matching* techniques. Apart from live synchronization, triggers can also be applied to detect and react upon complex details, e.g. on-the-fly well-formedness checking in domain-specific visual languages.

Experience with designing live transformations has shown that the solution introduced in [3] is too restrictive. It only allows the detection of a single pattern match (dis)appearance; neither simultaneous match set changes nor static context can be expressed to restrict the application of the trigger. This paper aims to introduce a significantly more general formalism for specifying graph triggers.

## II. Background

The central concept of GT is the notion of graph patterns, which are basically small graphs. Pattern matching is the (computationally complex) process of identifying subgraphs in the model that correspond to the pattern. More formally, the pattern contains a set of *pattern variables* with some

constraints attached to them; a pattern match is a mapping of all pattern variables to model elements so that the image of the variables observe all constraints. The most important constraints are entity constraints stating that a variable be a node of a certain type, and relation constraints stating that a variable be an edge of a certain type, connecting two given variables. Some advanced formalisms, like the pattern language [4] of the tool VIATRA2 may permit disjunction and negation of constraints, as well as equality and inequality, attribute constraints, or *pattern composition*. The latter means patterns *call*ing each other; a pattern call constraint may prescribe that given pattern variables be mapped into a match of a given graph pattern. Pattern composition facilitates reusing common patterns, improves expressiveness and pattern matching performance in some cases, and may even be used recursively under certain circumstances.

As an illustration, a GT rule of a simplified Object-Relational Mapping (ORM), that maps object-oriented classes into database tables, is shown in Figure 1. The LHS pattern matches classes that do not have an associated table, and the RHS shows a table that is traced back to the same class. Executing the GT rule results in mapping an unmapped class to a table. Here the class node ($C$), the table ($T$) and the traceability edge ($r$) are pattern variables; their types (class, table, trace respectively) and configuration are constraints in both patterns, and the *NEG* box expresses negation. ORM can also be executed as a live transformation; if the GT rule is triggered whenever a match of the LHS appears, new classes will automatically be transformed into tables.

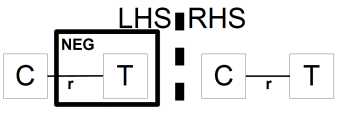



Figure 1: GT rule

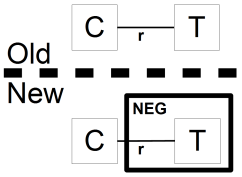



Figure 2: GGCP

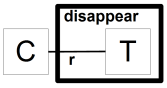



Figure 3: CP

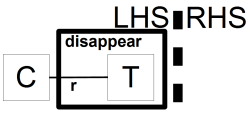



Figure 4: CGT

The aim to execute transformations without the costly re-evaluation of unchanged parts of the evolving source model is called source incrementality. Source incrementality can be achieved by employing incremental pattern matching techniques; for example, the RETE [5] incremental algorithm was used in [6]. The central idea of incremental pattern matching is that occurrences of a pattern are readily available at any time, and they are incrementally updated whenever changes are made. As pattern occurrences are stored, they can be retrieved in constant time – excluding the linear cost induced by the size of the result set itself –, making pattern matching a very efficient process. There are two important drawbacks; one of them is the increased memory consumption due to the stored occurrence sets. Additionally, these stored result sets have to be continuously maintained whenever the model is changed, causing an overhead on model manipulation. Nevertheless, benchmarks [7] and practice have shown that incremental pattern matching can improve performance or scalability by up to several orders of magnitude in certain scenarios. Moreover, incremental pattern matching leads to easy discovery of appearing or disappearing pattern matches, thus it can be used to efficiently implement graph triggers.

## III. General Graph Comparison Patterns

Triggers are essentially meant to make sense of the difference of two snapshots of the model (e.g. the state before and after a transaction), and to detect a high-level concept of events. The power of conventional graph triggers is restricted because their triggering condition (and LHS) does not have full access to the model in the two snapshots, but only to one element of the change set of pattern matches. For instance, if the ORM example needs to be a *bidirectional* live transformation, then new classes should be transformed into tables, while the deletion of the table should result in the deletion of the class; and likewise for mapping tables back to classes. The conventional trigger formalism cannot dif-

ferentiate between these cases without auxiliary structures; the previous example trigger would simply recreate a deleted table. To remedy this issue, I introduce the concept of *General Graph Comparison Pattern* (GGCP), which expresses the changes and unchanged parts between two versions of the graph, in graph pattern form.

**Definition.** Like conventional patterns, GGCPs also contain pattern variables that are to be mapped to graph elements. Each GGCP contains two sets of constraints, permitting all mentioned constraint types (including calling conventional patterns); one set should be satisfied in the old model, while the other is valid for the new model. As a special constraint type not belonging to either set, GGCPs can call each other. All four kinds of constraints can be the subject of disjunction or negation.

**Semantics.** A match is a constraint-satisfying mapping from GGCP variables to the union of graph elements present in the old and new model. The identity semantics are the following: elements existing both in the old and new model preserve their identity, i.e. a single variable will represent the same element in both snapshots for the purposes of the old and new constraint set. For elements that were created or deleted, and therefore only exist in one of the snapshots, basic constraints will only evaluate to true in one of the sets; their non-existence in the other snapshot can be explicitly checked using the negation of a sufficiently general constraint in the corresponding set. GGCP composition has the usual semantics.

**Discussion.** This formalism is powerful in the sense that it has unrestricted access to both the state before and the state after the change, with the full expressiveness of graph patterns. As a demonstration, Figure 2 shows a GGCP for bidirectional ORM, that detects deleted tables (in order to delete the class), but excludes the creation of a class. The downside is that compared to the less powerful graph trigger formalism, GGCPs capture changes on a low level of abstraction, therefore they are not as intuitive as more difficult to specify for a range of practical applications. An additional problem is that it is not apparent how an implementation can efficiently recognize GGCPs, without actually storing an archive copy of the old state of the graph, which would be costly both in terms of memory consumption and runtime performance.

## IV. Change Patterns

To solve the limitations of conventional graph triggers and GGCPs, I propose the formalism of *Change Pattern*s (CP), inspired by Event-Condition-Action systems (e.g [8]), having access both to the high-level dynamic events happening between states and their static context in the new state, but without the costly direct access to the old snapshot. Thus CPs are high-level trigger guards, that are easy to specify, efficient to implement and powerful at the same time.

**Definition.** CPs also capture changes between two snapshots of a model. Each CP contains variables (with the same identity semantics as GGCPs) and one set of conventional constraints (including regular pattern calls) that are valid in the new state, to capture the static context of dynamic events. This set of constraints is extended by event queries, which consist of a sign (appearance / disappearance) and a pattern call, each of them similar to a conventional graph trigger guard. Finally, CPs are also allowed to call other CPs. All three kinds of constraints can be the subject of disjunction or negation.

**Semantics.** A match is a constraint-satisfying mapping from CP variables to the union of graph elements present in the old and new model. Conventional constraints have to be valid at the new state of the graph; they are always false for deleted elements, which can be explicitly detected using negation. Appearance events are satisfied iff the variables are mapped into a match of the called pattern in the new state that was not valid in the old snapshot. Conversely, disappearance events map the variables into matches in the old snapshot that are no longer valid in the new one. CP composition has the usual semantics.

**Discussion.** I argue that due to the higher granularity of capturing changes as appearance or disappearance of complex patterns, CPs are more intuitive and easier to specify than GGCPs for a vast

range of practical cases. Furthermore, they retain the efficiency of conventional graph triggers in most cases, as there is no need to keep an archive copy of the old model; it is sufficient to remember the delta of the match set of involved patterns. The latter task is easily accomplished by the incremental pattern matcher; as pattern matches can be rare and typically there is only a small number of appearing / disappearing matches as a result of one transaction, the associated costs are expected to be small.

It is worth noting that in addition to these these advantages in typical cases, CPs actually have the same expressive power as GGCPs. Basically GGCPs impose logical connectives (conjunction, disjunction, negation) on asserting (conventional) pattern constraints to be true only in the new state, only in the old state, in both states, or in neither one. As the constraint can be wrapped into a graph pattern and thus usable in event queries, CPs can also distinguish these four cases, respectively by imposing the appearance of the constraint, the disappearance, the presence in in the new state and negating the appearance, and negating both the presence in the new state and the disappearance. Since logical connectives are also available in CPs and GGCP calls can be translated into CP calls by induction, there is an equivalent CP for each GGCP. As a demonstration, Figure 3 shows the CP equivalent of Figure 2.

## V. Contextual Triggers

Triggers defined with a CP as a guard (instead of a conventional pattern with an appearance / disappearance sign) are *Contextual Graph Triggers* (CGT). For instance, the CP on Figure 3 and an empty RHS form a CGT that propagates the deletion of tables to the associated classes (Figure 4). CGTs are more expressive than the graph triggers introduced in [3], as detecting events (match set changes) can be extended with the context of the change, consisting of a static pattern or even simultaneously occurring events. With exactly one event query and no static context, contextual triggers degenerate into conventional ones. With no event query (just a static pattern), contextual triggers degenerate into regular GT rules, that are applicable based on the current state alone, regardless of history.

**Conclusion.** The newly introduced CGT formalism unifies the advantages (expressiveness, ease of use, efficiency) of several approaches to achieve live transformation. Future work includes dealing with multiple simultaneous trigger activations (e.g. with priorities, conflict resolution), and finding a more direct approach to detecting attribute changes.

## References


[1] H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, Eds., *Handbook on Graph Grammars and Computing by Graph Transformation*, vol. 2: Applications, Languages and Tools, World Scientific, 1999.

[2] D. Hearnden, M. Lawley, and K. Raymond, "Incremental Model Transformation for the Evolution of Model-Driven Systems," in *Proc. of 9th International Conference on Model Driven Engineering Languages and Systems (MODELS 2006)*, vol. 4199 of *LNCS*, pp. 321–335, Heidelberg, Germany, 2006. Springer Berlin.

[3] I. Ráth, G. Bergmann, A. Ökrös, and D. Varró, "Live model transformations driven by incremental pattern matching," in *Proceedings of 1st International Conference on Model Transformation*, LNCS. Springer, 2008.

[4] A. Balogh and D. Varró, "Advanced model transformation language constructs in the VIATRA2 framework," in *ACM Symposium on Applied Computing — Model Transformation Track (SAC 2006)*, 2006.

[5] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," *Artificial Intelligence*, 19(1):17–37, September 1982.

[6] G. Bergmann, A. Ökrös, I. Ráth, D. Varró, and G. Varró, "Incremental pattern matching in the VIATRA model transformation system," in *Graph and Model Transformation (GraMoT)*, G. Karsai and G. Taentzer, Eds. ACM, 2008.

[7] G. Bergmann, A. Horváth, I. Ráth, and D. Varró, "A benchmark evaluation of incremental pattern matching in graph transformation," in *International Conference on Graph Transformation*, 2008.

[8] J. J. Alferes, F. Banti, and A. Brogi, "An event-condition-action logic programming language," in *JELIA*, M. F. et al., Ed., vol. 4160 of *Lecture Notes in Computer Science*, pp. 29–42. Springer, 2006.

# STATIC ANALYSIS OF MODEL TRANSFORMATIONS

**Zoltán UJHELYI**
**Advisor: Dániel VARRÓ**

## I. Introduction

Model transformations have a crucial role in the model-driven development [1] processes as they form the basis to derive source code from high level descriptions. Usually complex model transformations are captured by transformation programs.

As these programs grow in size ensuring their correctness becomes increasingly difficult, nonetheless it is required as errors in these programs can propagate into the developed application.

Methods for ensuring correctness of computer programs such as *static analysis* are applicable for transformation programs as well. Static analysis represents a set of techniques for computing different properties of programs without their execution. It is extensively used both in compiler optimization and program verification.

As computing the properties of the program can be infeasible to calculate - or even undecidable, - static analysis tries to calculate an approximation of these properties. In case of using an over-approximation, it can be guaranteed that no *bugs are missed*, similarly using an underapproximation may imply that no *spurious warnings* (an error message about a bug not present in the application) are emitted.

A widely used static analysis methods is abstract interpretation [2], that is based on the concept of abstract domains. The elements of the abstract domains categorize the concrete values (as present in the code) by a selected property. Using abstract interpretation the analyzer evaluates the program on these abstract domains to provide an approximation of the behavior.

## II. Overview of the Approach

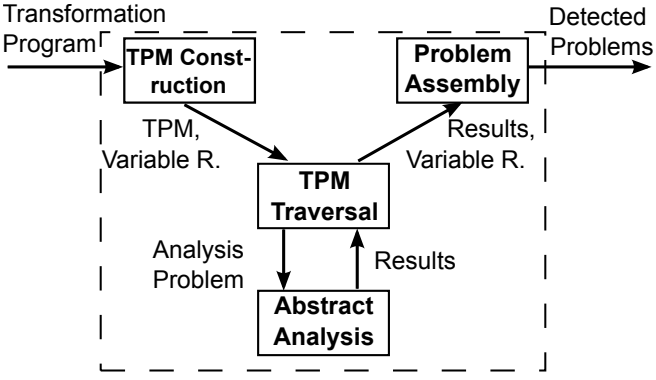



Figure 1: Overview of the Analysis

We implemented a static type checker component for the VIATRA2 model transformation system [3] that maps the variables of the transformation programs to their types, and validates it. The component is implemented in a generic way in order to support the analysis of other properties. For type checking every variable of the program has been mapped to the abstract domain of types, and the type information have been inferred following every possible execution path.

The static analysis process consists of three separate steps: (1) a Transformation Program Model (TPM) is built for storing an abstract representation of the program, (2) the model is analyzed using an abstract analysis tool and (3) the problems are connected. Figure 1 shows an overview of the static analysis process.

The TPM model is created to represent the transformation program in abstract domains - e.g. in case of type checking every variable is replaced with its type.

The main part of the static analysis consists of traversing the TPM model, and incrementally building and evaluating an analysis problem. The analysis problem of the type checking task is implemented as

a constraint satisfaction problem [4], where program variables are mapped to constraint variables, and type information is represented in the domains of CSP variables.

In order to give useful error messages to the transformation developer, the analysis results have to be back-annotated to the transformation program. This is achieved by the incremental building of the CSP from the TPM, which allows to pin-point the faulty variable. The back-annotated error messages fit into the following categories: (1) *analysis problems* mean the abstract analysis fails, (2) *inconsistencies* happen when multiple analysis iterations report contradictory values and (3) *traversal problems* indicate that the traversal could not finish the analysis properly.

The whole analysis approach is described in more details in [5].

## III. Enhancing the Static Analysis Framework

As with all static analysis technique the applicability of the static analysis framework to real life scenarios is a crucial question, and mainly relates to the performance aspect. My framework traverses every execution path separately, and the number of execution paths can increase exponentially with the size of the transformation programs.

Therefore significant performance increase can be achieved by a *modularized traversal*, that divides the transformation program into smaller, verifiable parts.

As the parts of the transformation program interact with each other, the relation between the parts have to be described. For this reason a *contract* [6] is attached to every validated part that describes a part of the static analysis results, that are relevant to the interaction between the different parts - e.g. in case of type checking the inferred types of the parameters are added to the contract. After the contract is created, every use of the program part can be replaced by its contract.

## IV. Evaluation and Future Plans

Type errors are hard to detect manually as transformation programs can be executed without runtime errors, only the output differs from the expected outcome. The created static type checker tool could detect these type errors that makes it a usable addition to the VIATRA2 framework.

The performance of the tool using the modularized traversal is also acceptable: it was capable of type checking of a real-world transformation program [7] in a reasonable amount of time (a few minutes).

As for the future we are planning to extend the system with other validations such as *dead code analysis* to detect code segments that cannot be reached and *use-definition analysis* to detect use of uninitialized or deleted variables. Additionally we plan to evaluate the performance of the analysis using SAT solvers as the underlying abstract analysis tool.

## References


[1] Object Management Group, *Model Driven Architecture — A Technical Perspective*, September 2001, `http://www.omg.org`.

[2] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 238–252, Los Angeles, California, 1977. ACM Press, New York, NY.

[3] D. Varró and A. Balogh, "The model transformation language of the VIATRA2 framework," *Science of Computer Programming*, 68(3):214–234, October 2007.

[4] K. Apt, *Principles of Constraint Programming*, Cambridge University Press, 2003.

[5] Z. Ujhelyi, A. Horváth, and D. Varró, "A generic static analysis framework for model transformation programs," Technical Report TUB-TR-09-EE19, Budapest University of Technology and Economics, June 2009.

[6] B. Meyer, *Object-oriented software construction (2nd ed.)*, Prentice-Hall, Inc., 1997.

[7] M. Kovács, D. Varró, and L. Gönczy, "Formal Analysis of BPEL Workflows with Compensation by Model Checking," *IJCSSE*, 23(5), November 2008.

# AUTOMATIC REFINEMENT OF TRANSFORMATIONS BY EXAMPLES

**Gábor GUTA**
**Advisors: Wolfgang SCHREINER, Dániel VARRÓ**

## I. Introduction

Model driven software development (MDSD) is a promising method to deliver software more efficiently. In MDSD software is generated from models with the help of transformations. Current industrial MDSD technologies focus on the production of the software with the help of out-of-the-box transformations, but provide no efficient support for the development of new transformations [2]. This problem has been recently recognized by the research community leading to the paradigm of model transformation by example (MTBE). In MTBE the transformation is inferred from examples by various methods, but there are still several open questions [4].

Our goal is to provide automatic or semi-automatic inference of transformations according to an approximate solution of the transformation problem provided by the developer. This approach is extending the traditional MTBE approaches by having additional information from the approximate solution to guide the inference process. In this paper we describe the overall problem we are going to address in the near future. We also present our preliminary results on a simplified model of transformation, namely on finite string transducers.

## II. Problem statement

In a model transformation system we usually generate code $g$ from a specification $s$ by a given transformation $t$. The generated code is determined by the specification and the transformation $g=t(s)$. We may define an expected result of the transformation $g'$, which is produced from $g$ by modification $m$, i.e.: $g'=m(g)$. It seems an interesting research topic to investigate whether it is possible to modify the approximate transformation $t$ to the expected transformation $t'$ automatically to generate the modified code $g'$ from the original specification, i.e.: $g'=t'(s)$. With fixed $g$ and no constraints on the $t'$ the task is trivial, e.g. the algorithm may just learn to give the right answer to the specified example. In order to get a non-trivial solution, constraints (metrics of the generalisation properties of the algorithm) need to be introduced on $t'$.

Therefore we reformulate this problem in the following way: $S$ is a set of possible specifications. G is a set of generated codes where each element $g$ of $G$ has a corresponding element $s$ of $S$ such that $g=t(s)$. Take a small subset of $S$ and call it $S_k$. Call $G_k$ the subset of $G$, containing the elements of $G$ which correspond to the elements of $S_k$. Then we may introduce modification $m$ over the elements of $G_k$, the result of which will be the members of the set $G_k'$. Thus the reformulated question is, whether it is possible to infer algorithmically an optimal transformation $t'$, which transforms the elements of $S_k$ to the corresponding $G_k'$ elements in such a way that a certain "optimality" criterion is satisfied. To express the optimality of the transformation we have to introduce metrics. These metrics cannot be computed by $t$ alone, but are also influenced by other characteristics related to the transformation, e.g. similarity of $t$ to $t'$.

## III. Our approach

The main difference of our approach from other model transformation by example approaches is that we require extra information in form of an approximate solution. Model transformation by example can be viewed as a transducer inference problem. (A transducer is an automaton with input

and output.) According to the theoretical results of the grammatical inference field we are aware of the negative results of algorithmic learning of even relatively simple computational models, e.g. inferring simple regular grammars [3]. We start our investigation with studying the properties of learning finite state string transducer modifications.

Given a finite state string transducer and an example string e.g. ({a,b},{x,y,w,z},{A,B},{A},{(A,a)->(A,x),(A,b)->(A,y)}) and "aabaab", our experimental setting to evaluate inference algorithms is the following:

- The transducer is executed using the example string as an input. The result of an execution is an output string and sequence of the executed transitions (trace), e.g.: “xxyxxy” and [(A,a)->(A,x), (A, a)->(A,x), (A,b)->(A,y), (A,a)->(A,x), (A,a)->(A,x), (A,b)->(A,y)].
- Modifications are inserted into the output by us, e.g.: “xxyzxxyz” and a difference string is generated from the original and the modified output string (the difference string contains information from both versions of the output, as well as their relation) by a selected “diff” algorithm. Example difference string: [x, x, y, +z, x, x, y, +z].
- The inference algorithm is executed with the given transducer, trace, and difference string generated during the previous step. The algorithm generates the new transducer, which is evaluated by us. Transitions of the inferred automaton: {(A,a)->(A,x), (A,b)->($New_1$,y), ($New_1$, λ)->(A, z)}, where λ denotes the empty symbol.

The basic idea behind the transducer learning algorithm is that it tries to modify the transducer transitions as little as possible at the same time making it capable to produce the expected modified output. The expected solution according to the human intuition is usually one of the many valid inferred solutions, to obtain this intuitive solution extensive tuning of the algorithm implementation is needed. This automatically raises the question how to formalize the intuitive expectation of the inferred solution in form of objective quality criteria. We can use simple product metrics [1] e.g.: number of new states, number of modified transition rules; or we can use constraints on the expected output e.g.: “If we introduce two equivalent changes in the output positions in which the same unique state transition occurs before the insertion positions, then the corresponding modifications of the transducer have to be equivalent.” According to our experience such constraints are more important than single metrics to specify the quality of the inferred transducer. We also noted that the information from the preceding steps, like the selection of the diff algorithm, also heavily affects the quality of the solution.

## IV. Conclusion

Finite state string transducers appear as a promising computational model to study properties of transducer modification learning algorithms. We are going to investigate computational models to which realistic examples can be mapped as well, e.g. tree transformations.

## References


[1] N. E. Fenton and S. L. Pfleeger: *Software Metrics: a Rigorous and Practical Approach 2nd*, PWS Publishing, 1998.

[2] G. Guta, W. Schreiner, and D. Draheim: “A Lightweight MDSD Process Applied in Small Projects,” in *Proc. of the 35th Euromicro Conference Software Engineering and Advanced Applications*, pp. 255–258, 2009.

[3] C. Higuera: “A bibliographical study of grammatical inference,” *Pattern Recognition*, 38(9): 13322–1348, 2005.

[4] D. Varró and Z. Balogh: “Automating model transformation by example using inductive logic programming,” in *Proc. of ACM Symposium on Applied Computing*, pp. 978–984, 2007.

# BPEL VERIFICATION: THE BACK-ANNOTATION PROBLEM

**Ábel HEGEDÜS**
**Advisor: Dániel VARRÓ**

## I. Introduction

Complex distributed systems including numerous business processes are widely employed nowadays. Business processes implemented in BPEL [1] (Business Process Execution Language) are often used for providing autonomous services. Since the quality of these processes are frequently critical for its users, their design-time verification is essential. Numerous approaches, techniques and tools were developed to support verification, usually by modeling the behaviour of BPEL processes using formal languages.

Modeling the BPEL process, however, is only part of the solution. The results of the verification carried out on the formal model have to be described using the formalism of BPEL. This reverse projection of information is known generally as the *back-annotation* problem.

Given that the verification results describe an execution of the BPEL process, modeling the structure and behaviour of BPEL (*static model*) is not sufficient for providing back-annotation support. Therefore a *dynamic model* is defined, which describes the actual state of the process (e.g. activities currently executed) together with a *simulation trace model*, to represent the information regarding the history of changes in the dynamic model.

In this paper, we describe how the simulation trace models can be defined for a given BPEL verification approach [2]. Furthermore a brief description of the back-annotation transformation implemented in the VIATRA2 framework [3] is given.

Several different simulation models have been proposed, for instance [4] uses models which include the dynamic behaviour information as well as the static and defines transformation rules for the simulation of these models. However, the dynamic model doesn't use any generic metamodel and the simulation trace is not stored. This approach was implemented in the AToM$^3$ [5] framework, which provides meta-modeling and model transformation support. [6] introduces change history models which are similar to simulation models although defined on a lower level. Model changes are recorded incrementally in order to enable model synchronization.

## II. BPEL modeling with Labeled Transition Systems

The BPEL standard does not define formal semantics for business processes. However, design-time verification of BPEL processes demands approaches which provide semantics through modeling business processes using formal languages. In the selected method, labeled transition systems defined in the Symbolic Analysis Laboratory (SAL) [7] were chosen for this purpose, and requirements against the BPEL processes are verified using model checking.

The SAL model of a given BPEL process is generated by a complex model transformation (BPEL2SAL). The various elements of the BPEL process are represented as stateful SAL variables and the process execution is modeled with transitions that alter the state of variables if the transition system is in a given state. Note that the BPEL2SAL transformation abstracts the behaviour of processes [2]. A sequence of state changes (transition firings) correspond with the execution of the BPEL process. While model checking of the SAL system may return a counter-example, deriving the corresponding BPEL execution is a non-trivial back-annotation problem (see Fig. 1).

## III. Back-annotation of the counter-example

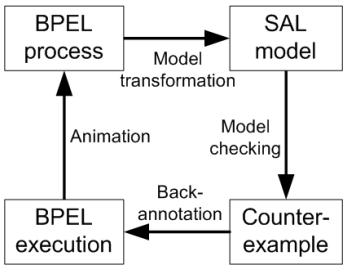



Figure 1: Approach

The BPEL execution (target trace) can be derived automatically from the SAL counter-example (source trace) by means of a model transformation, though this transformation is not the reverse of the original BPEL2SAL one. Instead of altering the static models, runtime information is represented in dynamic models and the traces are modeled separately using simulation metamodels. The SAL counter-example is modeled as a series of *steps* which represent the *firing transition* and the *variable assignments* it invokes. The assignments refer to *runtime variables* (in the dynamic model) that store the actual value of a variable in the static model and the values of the variable before and after the assignment. By storing the earlier value, backwards stepping in the trace is possible.

The simulation metamodel for the BPEL execution is similar to the SAL semantics given that the purpose of the original BPEL2SAL transformation was to model BPEL semantics as a transition system. Thus the various elements of the BPEL process are modeled as stateful entities and the steps of the execution represent state changes of the different entities. It is important to note that the simulation metamodels did not exist before and were devised in order to make the back-annotation possible.

The back-annotation transformation is implemented as an interface in the sense that it provides the following functions for handling the BPEL trace: (a) *initialize* for creating the dynamic BPEL model and the empty trace; (b) *forward step* for updating the dynamic model according to the next step in the trace, the step is generated based on the corresponding SAL trace step if it does not exist yet; (c) *backwards step* for reverting the dynamic model to the state before the actual step; and (d) *reset* for returning to the start of the trace and the initial state of the dynamic model. After the execution of each function, the state changes of the BPEL elements are exported so that they can be used outside the model transformation framework (e.g. to drive the animation of the BPEL process).

## IV. Conclusion

In this paper, we presented the design-time verification of BPEL processes as a challenging problem where automated back-annotation is essential. In order to provide transformation-based back-annotation of the counter-example returned by model checking, we defined dynamic and simulation trace models for both SAL and BPEL. The implemented tranformation can be used for animating the execution of the BPEL process.

Current research directions include exploring the possibilities of generalizing the simulation trace model and creating a domain-independent simulation interface providing similar functions as the described back-annotation transformation.

## References

[1] OASIS, "Web services business process execution language version 2.0 (OASIS standard)," 2007.

[2] M. Kovács, D. Varró, and L. Gönczy, "Formal Analysis of BPEL Workflows with Compensation by Model Checking," *IJCSSE*, 23(5), November 2008.

[3] Fault Tolerant System Research Group, "VIsual Automated model TRAnsformations," http://www.eclipse.org/gmt/VIATRA2/.

[4] J. de Lara, "Meta-modelling and graph transformation for the simulation of systems," *Bulletin of the EATCS*, 81, 2003.

[5] J. de Lara and H. Vangheluwe, "AToM$^3$: A tool for multi-formalism and meta-modelling," in *FASE*, 2002.

[6] I. Ráth, G. Varró, and D. Varró, "Change-driven model transformations," in *Proc. of MODELS'09, CM/IEEE 12th International Conference On Model Driven Engineering Languages And Systems*, 2009, Accepted.

[7] N. Shankar, "Symbolic Analysis of Transition Systems," in *ASM 2000*, Y. Gurevich, P. W. Kutter, M. Odersky, and L. Thiele, Eds., number 1912 in LNCS, pp. 287–302, Monte Verità, Switzerland, mar 2000. Springer-Verlag.

# AUTOMATED ROBUSTNESS TEST GENERATION USING OCL CONSTRAINTS

**János OLÁH**
**Advisor: István MAJZIK**

## I. Introduction

The use of different engineering models in software development is prevalent. As models are getting more detailed and sophisticated, they describe the system more precisely. Creating these models takes more effort from software developers, hence the demand to reuse these detailed models in every step of a software development process is a common goal.

Testing is a primary software verification technique used by developers today, but unfortunately a formal comprehensive method does not exist. Model-based testing is a state-of-the-art way to address the problem of software testing (a good overview is presented in [3]). It appeared first by using precise mathematical models (e.g. FSM, LTS) to determine optimal test-cases (optimal in a sense of best coverage in unit time). Several solutions exist how to transform other engineering models into these precise formal models in order to allow automated test generation.

Other potential possibilities emerged with the introduction of the design by contract (DBC) approach (e.g. JML [4]). The principal idea behind DBC in object oriented systems is the following. A class and its clients have a contract with each other. The clients must guarantee certain conditions, before calling a method (preconditions), and the class must ensure certain conditions that will hold after the method call (postconditions). In DBC, these contracts are mapped to executable code, thus any runtime violation can be detected immediately. In case of JML, an automated test generation based on JML specifications is presented in [5]. Preconditions are used to identify the allowed ranges and boundary values of input parameters of method calls, while the postconditions are used to evaluate the results of the call, forming a test oracle. (Test oracles are separate evaluation units for go/no go test.)

Test generation based on constraint languages (e.g. Object Constraint Language [1]) is also a promising approach. OCL is a formal language used to describe constraints on UML models. The OCL expressions may be invariant conditions that must hold for the system or multiplicity constraints over objects described in a model. OCL expressions are used to specify pre- and postconditions of method calls as well. This paper describes an idea aiming at robustness testing of a software based on UML models, extended by OCL constraints.

Robustness is the degree to which the system operates correctly in the presence of exceptional inputs and stressful conditions. In case of robustness testing, test cases contain unexpected input parameters for the system under test and it is examined whether the response of the system is acceptable (proper error codes are returned without crash, restart, omission). Examining results obtained from such experiments, robustness of software under test can be estimated.

## II. Information from OCL Constraints

Using UML models in test generation, all the necessary information to call an operation with appropriate parameters can be extracted. By using OCL constraints, all additional information can be extracted from the model, like the acceptable range of a parameter's value, and boundary values as well.

For example, if the range for an input integer value $i$ is defined as positive and less then 100, possible values for robustness testing are obviously $i < 0$ or $i > 100$.

OCL postconditions specify what have to be guaranteed after the operation returns. Postconditions

not only define a range for the return value, but can determine several other conditions that must hold after the operation returns. In case of test generation, postconditions carry the necessary information to assess the results of operation calls.

Using the abstract syntax tree (AST) of the OCL conditions, it is possible to separate the different conditions from each other along $and$ and $or$ keywords, and check each subtree separately. Then applying Depth First Search, parameters can be evaluated according to their operators. This strategy is used both in case of preconditions (to extract input parameter values) and postconditions (to check return values).

## III. Tool Support

The tool supporting test generation based on OCL constraints is currently under development. It is implemented as an Eclipse plug-in, hence it is able to work on UML models created within Eclipse. The tool works on applications written in Java, since it uses Javassist [2] byte code manipulator to load classes and execute operation calls.

With the help of the Javassist library, the plug-in is able to find the class files containing the operations to test (from the set of appointed class files). The plug-in has a blank $TestExecutor$ class file, which is modified by reflective programming. This modification is done in order to create and load the objects to test, and call their operations with the appropriate parameters (see Figure 1). These latter steps are hidden from the user, executed automatically in the background.

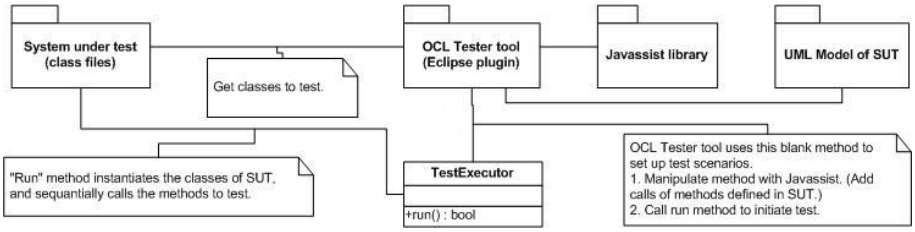



Figure 1: Test method instrumentation

## IV. Conclusion and Future Work

In this paper one possible use of OCL constraints in robustness testing was introduced. The tool using this approach is under development, supporting a subset of the OCL. Further research is needed to refine the selection of input parameters for the operations (combining exceptional inputs with random values, or using fitness functions for the selection), and to evaluate postconditions automatically. A promising complementary approach is to combine robustness testing supported by the tool with mutation of execution scenarios specified in the model.

## References

[1] O. M. G. Inc., *Object Constraint Language – OMG Available Specification Version 2.0*, May 2006, URL: http://www.omg.org/cgi-bin/doc?formal/2006-05-01.

[2] S. Chiba, "Javassist: Java bytecode engineering made simple," *Java Developer's Journal*, Jan. 2004.

[3] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, Eds., *Model-Based Testing of Reactive Systems*, Springer Berlin/Heidelberg, 2005.

[4] G. T. Leavens and Y. Cheon, *Design by Contract with JML*, Sept. 2006, URL: http://www.eecs.ucf.edu/ leavens/JML//jmldbc.pdf.

[5] F. Bouquet, F. Dadeau, and B. Legeard, "Automated boundary test generation from JML specifications," *Lecture Notes in Computer Science*, pp. 428–443, 2006, URL: http://www.springerlink.com/content/54q36j0r1362hnu5/.

# DEPENDENCY IDENTIFICATION IN SYSTEM MANAGEMENT

**István SZOMBATH**
**Advisor: András PATARICZA**

## I. Dependencies in IT

Today's business-driven IT infrastructures deliver complex services that are exploiting wide variety of IT resources. Thus the delivered business services require resources to operate as intended, for instance to provide appropriate response time to the end users. Dependencies between the provided services and the used resources can be static or dynamic. Traditional static dependencies are hard coded, thus they are created during design time. It is easier to discover them, since they does not saturate as often however this approach is not favorable, because it's inefficient resource usage. A more reusable approach is when a predefined topology is created or assumed during design time and resource instance binding is created during deployment. With dynamic resource allocation resource type deployment patterns are created during design time, the actual dependencies are bind (and may change) during runtime. In this way with dynamic resource allocation a resource friendly workload-driven (re)deployment approach can be attained.

Many approaches and applications exist to identify IT infrastructure dependencies. From the discovered dependencies a dependency map of the infrastructure can be reconstructed. An up to date topology map is indispensible to actuate a proper system management since dependability, performance and even security management requires information both on the structure and on the state of the supervised infrastructure. The first category of information on the structure on the system under control becomes more and more important in the point of view of practical system management due to the widespread use of adaptive architectures like those in autonomic grid and self healing systems, in which structural changes form an essential part of their operation.

The mined information from the dependency map that is a prerequisite of system management processes are impact and root cause analysis. That means for instance the identification of affected component upon a resource failure. Fast and accurate impact and root cause analysis has more and more importance because of the new emerging technologies like:

- diagnosis of faulty components,
- reconfiguration based fault tolerance,
- configuration consolidation,
- regrouping in green computing.

## II. Challenges and Approaches

### *A Data mining and machine learning*

Dependency detection plays crucial role in many fields, including supply chains, web services, network components, and even in bioinformatics. A dependency map can be represented with a directed graph, where the nodes are the components and the directed edges are the dependencies between the components. A common approach is to reconstruct the system model from observation i.e. from transitions and extract the dependency map from the model. In this way the problem shows similarities with workflow model reconstruction and state machine reconstruction, however there are differences as well. In process model mining it is assumed that a global template exists, but for some other applications, such as DNA and computer network analysis, there is usually no nontrivial order that can be expected globally. Thus data mining techniques like clustering, sequence and graph

mining that are able to perform well in noisy environment or even capable of dealing with inconsistency are preferable [1].

*B Flow based reconstruction of topology and dependency discovery*

Wide variety of dependency detection techniques exist. A favorable approach is to observe the network communication (e.g. with NetFlow), since it is not intrusive, no credentials are needed for the most observed servers and components, and it generates minimal extra workload on the system. A flow record is typically contains information about destination and source IP addresses and ports, send bytes and a timestamp. Fig. 1a shows a reconstructed network topology based on flow records. From the communication map of the network components dependencies can be mined out.

A common approach to discover dependencies is to identify correlated events. For instance an application server is (almost) always requested after a web server request. To find correlated relationships flow timings [4, 9] or data mining techniques [10] can also be used. A bit different approach but also dealing with correlated flows and time windows is Sherlock [12]. Their approaches and usability differs from each other a bit, although they are common in searching correlated events, like two flows happening very often within a given time interval.

Searching for patterns and classifying network traffic [8, 11] based on pattern matching not differs much from the previous approaches. BLINC is a promising research in this field. BLINC and tries to match typical infrastructure patterns (for instance Fig. 1b represents 3 tier architecture) to the infrastructure map. If a matching is found a set of flows has been classified. The drawback of this approach is that the patterns are created manually.

*C Updating the model and potentiality for real time analysis*

Synchronizing the models of the control system and the system under supervision has fundamental importance, since with false and obsolete dependencies proper impact and root cause analysis cannot be carried out. In a rapidly changing environment it is vital to track changes. Majority of existing dependency detection methods can deal with or reconstruct static models only, accordingly they are unable to catch such important affects as dynamic resource allocation and task migration and demand driven interactions (like those in SOA). An event driven change management has to be elaborated, that exploits the locality of the changes in typical dynamic structures, thus the dependency model is updated incrementally [6]. In this way upon model changes only the affected items will be updated.

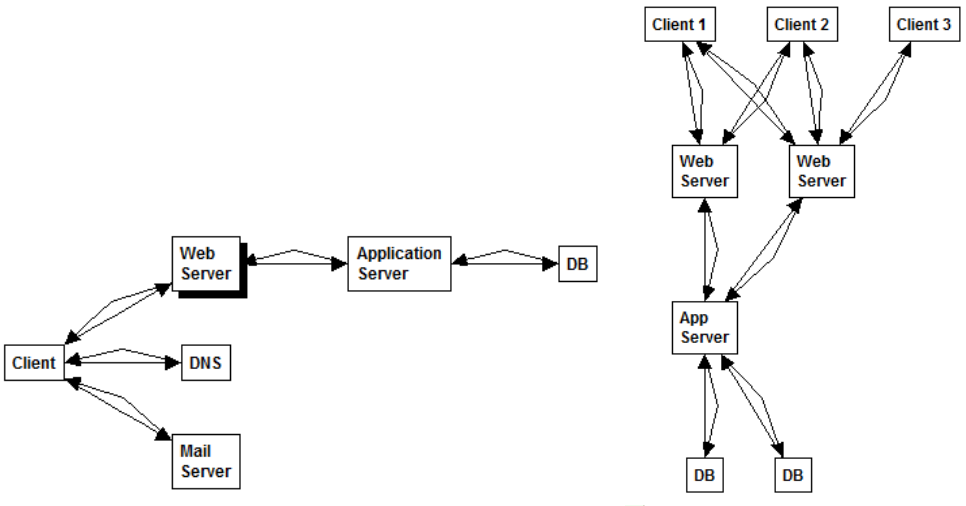



Figure 1a (left): Direct dependencies between IT infrastructure components.
Figure 1b (right): 3 tier architecture template.

## III. Identification of communication patterns

It is assumed that the automatic creation of typical business patterns is possible, because every enterprise IT infrastructure typically provides similar functionality like authentication, content sharing, business logic and data storage, and furthermore these functionalities are provided by deploying reusable templates based on best practices. The method presented here follows the idea of automatic pattern creation (possible pattern learning) and pattern matching principle. The estimation is that with graph clustering modules or clusters can be created that represents business service patterns. It is assumed that it is possible to (1) identify and (2) track the existence of typical high level patterns and (3, not presented here) mine partial orders from sequence data of subgraphs. It is expected that partial order mining of subgraphs is not as complex as in the whole dataset and most dependency information is not lost during the data divide.

Workflow for identifying service dependency patterns is as follows:

- typical high level service patterns are collected (e.g. with graph clustering);
- transitions of the system (e.g. flow records) are fed to the pattern matcher;
- upon new match the subgraph is labeled and the information is forwarded;
- the incremental and live framework ensures that upon change the model and the matched subgraphs are updated accordingly without rebuilding or recalculating the whole model space.

Automatic generation of patterns is an important step toward automated dependency discovery, since manual patterns may help verify the approach; it takes a lot of human interaction to create all relevant patterns. In this article a proof of concept is presented that shows the possibility to create patterns automatically using graph clustering.

Graph clustering identifies a group of nodes where the intra connectivity is higher that connection with other group of nodes. In this way a cluster consists of nodes that show strong connection with each other. With every clustering method, the distance or in this case the weight function needs to be designed carefully.

LinkLand clustering method is used from ModuLand method family [13]. The reason for that is:

- it is capable to identify modules, where a node may be part of more clusters, or so called modules,
- fast and scalable approach, it is possible to set the threshold limit that renders a node to a module,
- not just intra but also inter communication of modules can be analyzed

Small part of the clustered input data is shown on the Fig. 2. Clusters show highly connected subgraphs that may indicate typical business service patterns.

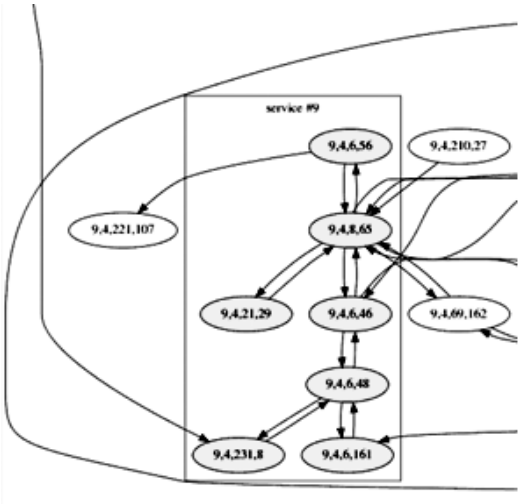



Figure 2: A cluster of a graph build from flow records.

## IV. Test Results and Conclusion

The method presented in this article is live or so called event driven and incremental. Event driven mechanism means that upon the detection of new input data (flow record) the model space is updated automatically and if new pattern match is found or a matched pattern becomes obsolete the appropriate application is informed automatically. Incremental approach means that upon change in the source model only the change is transformed and injected into the destination model, the entire model does not need a re-transformation. Incremental model transformation and pattern matching is ensured by the framework [6].

In this article the motivation to create and track dependency map using observations and a proof of concepts are presented. The demo implemented in a model transformation framework builds and maintains the IT infrastructure topology and is capable to mine higher semantics such as dependency information, strongly correlated services, and important services from flow records. Flow record can be fed into the demo as they are detected (run-time).

In addition the implementation may be used in a standardized system management environment as well, because it contains a service which is able to synchronize between the labeled graph model and the standardized infrastructure model stored in an enterprise class CMDB solution (IBM TADDM).

Another outcome is that it is proved that with graph clustering relevant infrastructure patterns can be identified thus the pattern creation can be aided by automatism.

Future work includes measures of time and memory consumptions to prove the scalability of the approach, and the refinement of the graph clustering, and it also has to be verified that it is possible to mine most relevant dependency using partial order mining in relevant subgraphs.

## References


[1] Guozhu Dong, Jian Pei, *Sequence Data Mining*, Springer, 2007

[2] Fischer, I.; Meinl, T., "*Graph based molecular data mining - an overview*," Systems, Man and Cybernetics, 2004 IEEE International Conference on, vol.5, no., pp.4578-4582

[3] Basu, Sujoy, Casati, Fabio, Daniel, Florian, *Toward Web Service Dependency Discovery for SOA Management*, SCC '08: Proceedings of the 2008 IEEE International Conference on Services Computing, 2008.

[4] Xu Chen, Ming Zhang, Zhuoqing Morley Mao, Paramvir Bahl, *Automating Network Application Dependency Discovery: Experiences, Limitations, and New Solutions*, *OSDI, pg. 117-130, 2008.*

[5] Stijn van Dongen, *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, May 2000.

[6] Ráth, I., Bergmann, G., Ökrös, A., and Varró, D. 2008. Live Model Transformations Driven by Incremental Pattern Matching. In *Proceedings of the 1st international Conference on theory and Practice of Model Transformations* (Zurich, Switzerland, July 01 - 02, 2008).

[7] Tolksdorf, R. 2003. A Dependency Markup Language for Web Services. In *Revised Papers From the Node 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems* (October 07 - 10, 2002).

[8] Marios Iliofotou, Prashanth Pappu, Michalis Faloutsos, Michael Mitzenmacher, Sumeet Singh, George Varghese, *Network Monitoring using Traffic Dispersion Graphs (TDGs)*, Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, pp. 315- 320, 2007.

[9] Andreas Kind, Dieter Gantenbein, Hiroaki Etoh, *Relationship Discovery with NetFlow to Enable Business-Driven IT Management*, 1st IEEE/IFIP Int.Workshop on Business- riven IT Management, 2006.

[10] Alexandru Caracas, Andreas Kind, Dieter Gantenbein, Stefan Fussenegger, Dimitrios Dechouniotis, *Mining Semantic Relations using NetFlow*, Third IEEE/IFIP International Workshop on Business-driven IT Management, 2008.

[11] Thomas Karagiannis,Konstantina Papagiannaki, Michalis Faloutsos, *BLINC: Multilevel Traffic Classification in the Dark*, ACM SIGCOMM Computer Communication Review, vol. 35, no. 4, pp. 229-240, October, 2005.

[12] Paramvir Bahl, Ranveer Chandra, Albert Greenberg, Srikanth Kandula, David A. Maltz, Ming Zhang, *Towards Highly Reliable Enterprise Network Services Via Inference of Multilevel Dependencies*, SIGCOMM, pp. 13-24, 2007.

[13] Szalay Máté, Kovács István, Palotai Robin, *Új eljáráscsalád gráfok pontjainak átfedő klaszterezésére biokémiai, szociológiai, valamint távközlési alkalmazásokkal*, OTDK 2009.

# ONTOLOGY BASED ASSESSMENT OF DEVELOPMENT PROCESSES

**Zoltán SZATMÁRI**
**Advisor: István MAJZIK**

## I. Introduction

Our everyday life depends on software to a considerable extent, this way the reduction of the risks of design and implementation faults is of utmost importance. Software development processes are more and more subject to regulations fixed in (general and domain-specific) standards that define criteria for the selection of proper development methods. Accordingly, if software is deployed in a critical environment then an independent assessment is needed to certify that its development process is compliant to the criteria stated in the related standard. The goal of this work is to support the assessment of development processes and toolchains by elaborating a formal verification technique that allows the automated checking of the compliance to standards. On the analogy of classical model checking (that is applied to examine whether a formal design model satisfies some temporal requirements) we represent the development process and tools in a *process model* by means of using ontologies and use a reasoner to check whether the criteria originating in the standard are satisfied.

This vision necessitates the solution of the following tasks. First we formalize the requirements (criteria) in standards that concern the selection of methods and tools. Then we define the modeling techniques to describe the relation and hierarchy of methods, the capabilities of tools, and the construction of (domain-specific) development processes. Finally we elaborate of techniques that check the compliance of concrete development processes (constructed by process designers) to the requirements.

## II. Ontology based modeling and model checking

Ontologies are widely used as knowledge management mechanism to capture knowledge about some specific domains. Ontology languages use concepts and relationships between these concepts as a logic model. The ontology languages have a good formal semantics and automatic reasoning algorithms. In order to automate the analysis of an ontology, reasoners are used that are based on the description logic formulation of an ontology.

Web Ontology Language (OWL) is a knowledge representation language specification published by the W3C. This XML based representation of ontologies is supported by the most important tools and is used in this work.

## III. Formalization of the requirements

Formalisation is a prerequisite of both formal verification and synthesis support. In this work the focus is on the development processes for safety critical applications, and the EN50128 standard [1] for railway applications is analyzed. This standard defines five safety integrity levels (SIL) for development processes and describes methods that can be applied during the process. For each development step the mandatory (M), highly recommended (HR), recommended (R) and not recommended (NR) methods are described in a tabular form.

The main challenges during the requirement formalisation are the following ones: The development methods are refined hierarchically, i.e., several high level methods are decomposed into alternative

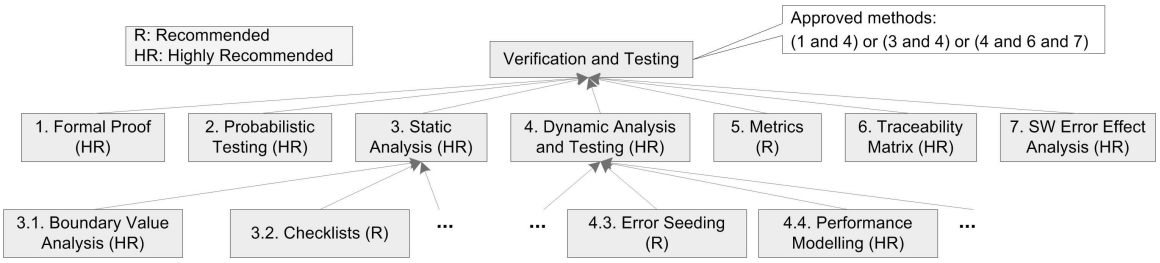



Figure 1: Verification and Testing methods for SIL4 (EN50128)

combinations of lower level ones (Fig. 1). Different requirements are described for each SIL in the standard. Accordingly, this introduces a new dimension into the requirement formalisation. Finally the sufficient conditions for every SIL are formulated using various combinations of the applied methods.

| Technique/Method | SIL1 | SIL2 | SIL3 | SIL4 |
|---|---|---|---|---|
| 1. Formal Proof | R | R | HR | HR |
| 2. Probabilistic Testing | R | R | HR | HR |
| 3. Static Analysis | HR | HR | HR | HR |
| 4. Dynamic Analysis and Testing | HR | HR | HR | HR |
| 5. Metrics | R | R | R | R |
| 6. Traceability Matrix | R | R | HR | HR |
| 7. SW Error Effect Analysis | R | R | HR | HR |

Figure 2: The Verification and Testing methods (EN50128)

In the following, the *Verification and Testing* step of the development process described in the EN50128 standard is presented as a small example in order to demonstrate the mentioned concepts. In Fig. 2 the recommendation level of some methods is shown. The combination of the required methods are expressed as follows: *„For Software Integrity Levels 3 and 4, the approved combinations of techniques shall be (1 and 4) or (3 and 4) or (4, 6 and 7)"*

During the requirement verification step one of the *sufficient conditions* should become true on the process model and none of the *„not recommend"* methods should be used in the process.

These requirements can be represented as Boolean expressions and if these expressions come true on the input process model, then this process model is standard compliant. Because of the challenges in the requirement description (e.g. hierarchical refinement of methods) ontologies are used to characterize the methods and the tasks in the input process models and finally reasoning is used to verify the requirements.

## IV. Modeling development processes

The (domain-specific) development process is formalised using a process model which describes the tasks, input and output artifacts, the roles and tools involved in the development process. There are several general purpose process modeling languages (e.g. BPEL, BPML).

In this work the OMG's Software Process Engineering Metamodel (SPEM) specification is used, because the focus of the SPEM is development projects and it is defined as a meta-model as well as a UML 2 Profile. The Eclipse Process Framework supports this specification and is proposed in our environment to model the processes. Using this framework the process designer can construct the specific development process, can assign the available tools to the tasks of the process, or can choose from available toolchain patterns.

In order to support the logical reasoning as model checking, the process is represented using an ontology. W3C's OWL-S ontology supports the description of service composition as well as business

processes. In the following the process description capability of this ontology will be used to describe the development processes [2].

The OWL-S ontology defines the *Process* concept that can be an *Atomic Process* or *Composite Process*. *Atomic processes* correspond to single steps of the development processes and composite processes are decomposable into other processes. Their decomposition can be specified by using control constructs such as *Sequence* and *Choice*.

The tasks of the process implement particular methods that can be classified by the standards, and based on this classification the assessment can be supported. A formal representation of the hierarchical structure of methods can be provided by defining a new ontology. Here concepts refer to the development methods and their relations include the refinement. The OWL-S ontology is specialized in order to support this method classification and this extension is called *methods ontology*. In this extension new concepts are defined as subclasses of the *Atomic Process* concept (e.g. Fault Tree Analysis, Probabilistic Testing.)

A simple example development process is shown in Fig. 3. Note that this development toolchain is compliant to the standard since the combination of Formal Proof (implemented by the SAL model checker) and the Symbolic Execution (which is a Static Analysis method implemented by the PolySpace tool) is a valid combination for SIL3 and SIL4.

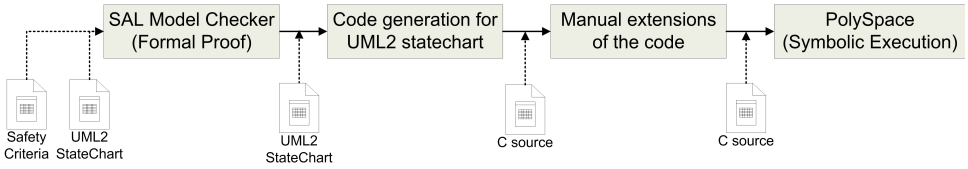



Figure 3: Sample development process

An additional step of the formalisation process is the construction of the tool repository. This repository is a collection of tools that can be used in a given company during the (construction of the) development processes. Each available tool is classified on the basis of the concepts defined in the *methods ontology* constructed in the previous step, i.e., for each tool the supported methods are given.

## V. Mapping SPEM to OWL-S

The required SPEM to OWL-S mapping is implemented using the VIATRA model transformation framework [3]. To use VIATRA the metamodel of the SPEM process modeling language and the metamodel of the $\mathcal{SHOIN}(\mathcal{D})$ ontology language is constructed in the VIATRA model space. After that the „SPEM to OWL-S" model-transformation (based on these metamodels) [4], an importer for the SPEM models and an exporter for OWL ontology format is implemented.

The SPEM model is constructed using the SPEM UML profile, so the UML metamodel (extended with UML profile support) is used to represent the input model of the transformation. The OWL metamodel is constructed based on the Ontology Definition Metamodel (ODM) developed by the University of Karlsruhe. [5]. This metamodel fits well into the Meta Object Facitlity (MOF) architecture and can be applied in the VIATRA model transformation framework.

## VI. The assessment toolchain

The assessment of development processes is implemented by an *assessment toolchain* in order to support automatic execution of the steps starting with the SPEM model transformation into ontology based models and finishing with the reasoning (Fig. 4)

First the process model is constructed by domain experts using the EPF Composer tool. This input model is transformed into an OWL-S based ontology then the used tools and atomic tasks are classified

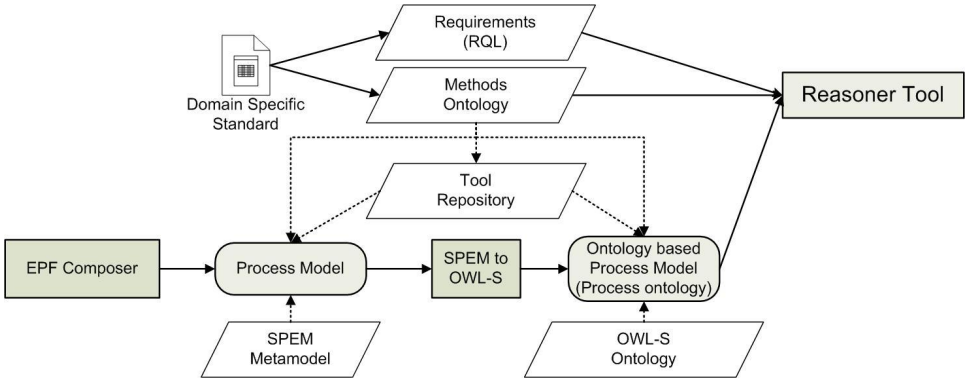



Figure 4: The assessment process

using the *methods ontology*. The output model is a process ontology. Finally, the standard conformance of the development process can be checked using the reasoner tool, that is executed on the process ontology.

According to the approach described above, all of the tasks, the tools and thus the development processes are characterized using the concepts represented in the ontology. Using the concepts defined in the ontology, the sufficient conditions for the selection of methods and the dependency on the safety integrity level should be represented as boolean expressions. These expressions can be described using ontology query language, for example the New RacerPro Query Language (nRQL). The query should check whether the required combination of the methods are included in the process model.

Accordingly, the standard conformance of the selection of methods and their supporting tools in the development process can be checked using an ontology reasoner.

## VII. Future work

During the standard based assessment of development toolchains not only the used methods are important. The standard specifies that safety arguments are required during the certification process. These safety arguments communicate the relationship between the evidence and objectives.

The arguments can be ordered into a *hierarchical breakdown structure*. There could be some parts of these arguments that are produced by tools in one step or by toolchains in multiple steps. The assessment process will be extended to support the construction of development processes by identifying missing safety arguments.

## References

[1] CENELEC, "En 50128: Railway applications - communication, signalling and processing systems - software for railway control and protection systems," URL: http://www.cenelec.eu.

[2] P. M. Anupriya, A. Ankolekar, M. Paolucci, and K. Sycara, "Towards a formal verification of OWL-S," in *In Fourth International Semantic Web Conference (ISWC 2005*.

[3] "VIATRA model transformation framework," URL: http://wiki.eclipse.org/VIATRA2.

[4] J. Shen, Y. Yang, C. Wan, and C. Zhu, "From BPEL4WS to OWL-S: Integrating e-business process descriptions," in *SCC '05: Proceedings of the 2005 IEEE International Conference on Services Computing*, pp. 181–190, Washington, DC, USA, 2005. IEEE Computer Society.

[5] S. Brockmans, P. Haase, P. Hitzler, and R. Studer, "A metamodel and UML profile for rule-extended OWL DL ontologies.," in *ESWC*, pp. 303–316, 2006.

# RECONFIGURABLE IMAGE PROCESSING PIPELINE

**Tamás RAIKOVICH**
**Advisor: Béla FEHÉR**

## I. Introduction

FPGA based hardware accelerators have become more and more widely used in different kind of applications. As compared to the ASICs, the advantage of the FPGA devices is their flexibility that arises from their programmable nature. In addition to this, the modern Xilinx Virtex FPGA series have a unique capability called partial reconfiguration [2]. During this process, the FPGA remains fully functional while a part of the device is reprogrammed. Usually, algorithms that require executing simple operations on large amount of data can be efficiently accelerated using FPGAs.

Many image processing algorithms belong to this group. Biological and biomedical experiments (for example microarray experiments, HCS or cellular microscopy) often result in large amount of image data that have to be processed in order to evaluate the experiment. There are different software applications available for this purpose, such as the open-source CellProfiler [1], which uses an algorithm pipeline to batch process cellular microscopy images. The mode of operation of the CellProfiler software served as a base for the hardware design.

This paper introduces a Virtex-5 FPGA based reconfigurable system for image processing purposes. The system consists of pipelines and each pipeline consists of several reconfigurable execution units. By using the partial reconfiguration capability of the FPGA, the pipeline stages can be quickly reconfigured with the required algorithms.

## II. The hardware design

### *A. Overview*

This design (Fig. 1.) is the improved and expanded version of the previous design described in [3]. The system is implemented on the ML506 board, which utilizes an XC5VSX50T FPGA optimized for memory-intensive and DSP applications. The communication, the partial reconfiguration and the image processing pipelines are controlled by the software running on the MicroBlaze processor. In the current design, the Ethernet interface is used to communicate with the PC. In case of a real hardware accelerator, a higher performance communication interface is required to provide the necessary bandwidth, such as PCI-Express, HyperTransport or FSB. The whole system runs at 125 MHz, except the ICAP (Internal Configuration Access Port) which requires 100 MHz clock.

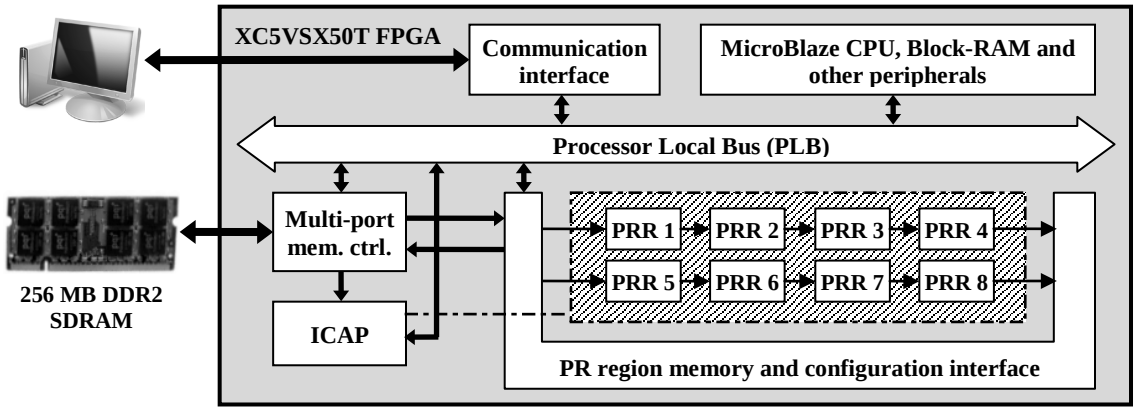



Figure 1: Block diagram of the system

### B. *Memory interface*

The memory interface greatly affects the performance of the system. Because of the limited number of the fast internal Block-RAMs, external memory is necessary for storing the large amount of image and configuration data. The ML506 board contains a 32M x 64 bit (256 MB) DDR2 SDRAM memory module, which is connected to the system through the multi-port memory controller.

Assuming 125 MHz system clock frequency, the theoretical peak bandwidth of the DDR2 SDRAM is 2000 MB/s. The reconfigurable modules have an 8-bit data interface, whose theoretical peak bandwidth is 125 MB/s. Comparing these values, it can be seen that maximum sixteen 8-bit data interfaces are recommended to connect the PR regions to the memory.

In this design, the memory interface unit has four 8-bit read and four 8-bit write ports for the reconfigurable modules. The read and the write transfers are scheduled using round-robin scheduling. The previously used XCL (Xilinx CacheLink) interface between the memory interface unit and the multi-port memory controller has been replaced with the NPI (Native Port Interface), which provides much better performance.

### C. *Reconfiguration (ICAP)*

The ICAP peripheral is necessary for the partial reconfiguration, it writes the configuration data into the internal configuration port of the FPGA. The new version of the ICAP peripheral is directly connected to the multi-port memory controller through the NPI interface.

## III. Image processing functions

In order to determine the resource requirements and the size of the PR regions, several image processing functions have to be examined. In this design, the cell recognition has been chosen to implement. This algorithm determines the borders of the cells on the cellular microscopy images. It executes different image processing tasks on the image data, which will be briefly discussed in the following sections. These functions can process 8-bit grayscale images of 2048 x 2048 pixels maximum size.

### A. *Noise removal*

The input images may contain noise that has to be removed before further processing. Using a 2D median filter [3], the “salt & pepper” noise can be efficiently reduced or removed from the images.

### B. *Histogram computation*

This stage determines the distribution of the pixel intensity values of the median filtered image. In case of 8-bit 2048 x 2048 grayscale images, the histogram computation requires a 256 x 23 bit RAM. At first, the RAM is filled with zeroes. Then the value addressed by the input pixel data is incremented in every step. After processing the last pixel, the RAM will contain the histogram.

### C. *Threshold value computation*

The histogram-based Otsu method is used to determine the threshold value for image binarization. The algorithm divides the pixels into two classes (foreground and background) then calculates the optimum threshold separating the two classes so that their $\sigma^2_{between}(t)$ between-class variance is maximal. After computing the initial values, the class probabilities ($P_{BG}$, $P_{FG}$) and the class means ($\mu_{BG}$, $\mu_{FG}$) can be updated recursively as each threshold value t is tested. P(t) is the value in the histogram memory at address t.

$$\sigma^2_{between}(t) = P_{BG}(t)P_{FG}(t)\left[\mu_{BG}(t) - \mu_{FG}(t)\right]^2 \tag{1}$$

$$P_{BG}(t+1) = P_{BG}(t) + P(t) \quad and \quad P_{FG}(t+1) = P_{FG}(t) - P(t) \tag{2}$$

$$\mu_{BG}(t+1) = \frac{\mu_{BG}(t)P_{BG}(t) + tP(t)}{P_{BG}(t+1)} \quad and \quad \mu_{FG}(t+1) = \frac{\mu_{FG}(t)P_{FG}(t) - tP(t)}{P_{FG}(t+1)} \tag{3}$$

The disadvantage of this algorithm is that it requires division, which cannot be easily implemented in FPGA devices.

### *D. Image binarization*

After computing the threshold value, each pixel of the median filtered image is tested. If the pixel intensity is lower than the threshold, it results a black binary image pixel (0). Otherwise the binary image pixel will be white (255).

### *E. Edge detection*

The 2D FIR filter [3] can detect the edges in the binary image when the Laplace operator matrix is used as the filter mask.

## IV. The reconfigurable regions

### *A. Size*

Table 1 shows the resource requirement of each image processing function. Because of the simplicity of the histogram computation and the image binarization, these are merged with the subsequent functions.

Table 1: Resource requirement and utilization

| **Image processing function** | **Cfg. file size (bytes)** | **Resource requirement and utilization** | | | |
|---|---|---|---|---|---|
| | | **LUT** | **Flip-flop** | **RAMB36** | **DSP48E** |
| Median filter | 71532 | 664 / 960 (70%) | 600 / 960 (63%) | 1 / 4 (25%) | 0 / 16 (0%) |
| Histogram + threshold | 71532 | TBD | TBD | TBD | TBD |
| Binarization + FIR filter | 71532 | 406 / 960 (43%) | 402 / 960 (42%) | 1 / 4 (25%) | 9 / 16 (57%) |

The height of the PR regions is determined by the smallest addressable configuration memory segment of Virtex-5 FPGAs, which is a 20 CLB-height frame. The XC5VSX50T device has only 6 frames in a column therefore the height of each PR region should be 1 frame. PR regions with different sizes have been tried. The result of the experiments was that the place and route operation failed when more than the 75% of the logic resources were utilized within a PR region. As a consequence, an optimal PR region contains 120 CLBs (960 LUTs and FFs), 4 Block-RAMs and 16 DSP48E slices. The complexity of the XC5VSX50T device allows placing eight reconfigurable regions into a design.

### *B. Interfaces and connections*

Fig. 2 shows the I/O interface of a PR region. Because the I/O signals routed into the reconfigurable modules through LUTs, their number should be minimized. Therefore, the PR modules have a simple data interface that consists of an 8-bit data bus and two control/status signals. The data ready signal is active when there is valid data on the data bus and the data acknowledge signal is active when the module is able to receive the next input. Every module requires a read (DATA_IN) and a write (DATA_OUT) data interface to communicate with the adjacent modules. Some functions (such as the image binarization) also require direct memory access, which is provided through the memory read (MEM_IN) and write (MEM_OUT) interfaces. The internal parameter registers can be written using the PAR_IN serial input line. The processing of the input data can be started by strobing the START input, the DONE output signals that the operation has been finished.

The connection of the PR regions can be seen in Fig. 3. In this design, the eight PR regions are arranged as two 4-stage pipelines. The adjacent pipeline stages are connected together through a FIFO, which helps to balance the variation of the execution speeds. The remaining ports of the memory interface units are available for the PR regions, a memory read and a memory write port can be connected to a stage in each pipeline. This allows dividing the 4-stage pipeline into two smaller parts.

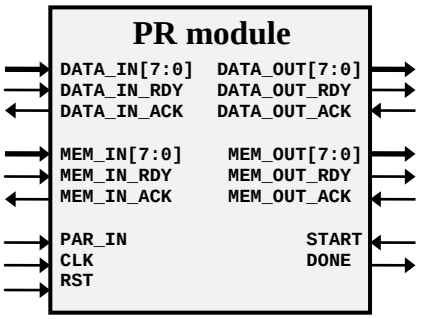



Figure 2: PR module interfaces

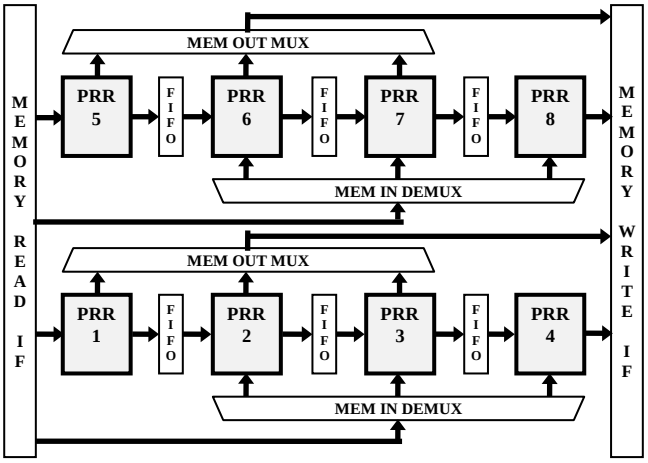



Figure 3: Connection of the PR regions

## V. Results

Except the threshold computation module, every part of the system has been completed and tested. Table 2 shows the reconfiguration and the filtering times of the filter modules. As for the "theoretical" columns of the table, it is assumed that every input data can be processed in one clock cycle. Contrary to the previous design [3], the "theoretical" and the "actual" values are almost the same because the redesigned memory interface provides the required bandwidth. Comparing the hardware and software filtering times, the FPGA design performs better.

At first, the PR modules are separately tested. In case of the 2D median filter, the speed-up is 70x. In case of the 2D FIR filter, the speed-up is 30x.

In the next test, the median and the FIR filters are connected together forming a 2-stage pipeline. In this case, the filtering time hasn't changed considerably. The new value is 2.107 ms, which means a 100x speed-up.

Comparing the reconfiguration and the filtering time, about 10 reconfigurations can be done while an image of size 512 x 512 pixels is processed by the filters.

Table 2: Reconfiguration and filtering times

| Module name | Reconfig. time @ 100 MHz | | Filtering time (512 x 512 image size) | | |
|---|---|---|---|---|---|
| | | | FPGA @ 125 MHz | | PC @ 2,8 GHz |
| | theoretical | actual | theoretical | actual | |
| Median filter | 0.178 ms | 0.18 ms | 2.097 ms | 2.1 ms | ~150 ms |
| FIR filter | 0.178 ms | 0.18 ms | 2.097 ms | 2.1 ms | ~60 ms |

## VI. Future work

Implementing more image processing functions is planned in the future. In the next version of the design, the currently used Ethernet communication interface should be replaced with PCI-Express interface, which can be found on the ML506 board.

## References

[1] CellProfiler cell image analysis software manual
URL: http://www.cellprofiler.org

[2] T. Raikovich, "*Dynamic Reconfiguration of FPGA devices*," in *Proc. of the 15th PhD Mini-Symposium*, pp. 72-73, Budapest, Hungary, February 4–5 2008.

[3] T. Raikovich, "*Image Processing Using Reconfigurable Hardware*," in *Proc. of the 16th PhD Mini-Symposium*, pp. 40-43, Budapest, Hungary, February 2 2009.

# AN FPGA-BASED, MASSIVELY PARALLEL MOLECULAR DOCKING MACHINE

**Imre PECHAN**
**Advisor: Béla FEHÉR**

## I. The Docking Problem

Molecular docking is an important problem of bioinformatics whose aim is to predict the binding geometry and binding energy of two molecules with computational methods. The two molecules are usually a receptor and a ligand. The receptor is a large protein molecule with a big pocket at its surface called active site, whose structure allows the binding of other compounds. The ligand is a much smaller molecule, whose binding energy and binding position inside the active site can be predicted by docking methods.

Molecular docking plays an important role in modern, computer-aided drug design. Many drug molecules work on the principle of competitive inhibition, which means that they block the activity of an enzyme (receptor) of the virus that causes the illness by binding to its active site. If the 3D structure of the enzyme is known, docking methods can help to find potential inhibitor molecules, thus improving the efficiency of the drug design process.

## II. Docking Algorithms

Docking algorithms usually consist of a scoring function and a search method. The scoring function models different chemical interactions, and can be used for determining the binding energy of a specific arrangement of the molecules. The aim of the algorithms is to find the global minimum of the scoring function, which corresponds to the energetically most favorable position of the molecules relative to each other. This role is filled by the search method, which is generally some kind of optimization algorithm often based on heuristics. During search the receptor is usually rigid and it is fixed in space, the degrees of freedom of the problem are the parameters that describe the position and orientation of the ligand relative to the receptor, and the torsional angles of the rotatable bonds inside the molecules.

### *A AutoDock*

There are dozens of software tools, which apply different scoring functions and search methods for solving the docking problem. One of them is AutoDock, which is a free, open-source, and therefore quite popular docking software.

AutoDock uses the following semi-empirical scoring function:

$$V = \sum_{i,j} \left[ W_{vdw} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) + W_{hb} \left( \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) + W_{el} \frac{q_i q_j}{\varepsilon(r_{ij}) r_{ij}} + W_{ds} (S_i V_j + S_j V_i) e^{-r_{ij}^2 / 2\sigma^2} \right].$$

The expression has to be summed over all non-bonded ($i$, $j$) atom pairs of the system, that is, the total free energy is the sum of energy components from pairwase atom-atom interactions. The expression consists of four terms, which represent the energy contribution of the van der Waals interaction, the hydrogen bonds, the electrostatic interaction and the desolvation effect, respectively. $A$, $B$, $C$, $D$, $S$ and $V$ are constants which depend on the types of atoms $i$ and $j$, $r$ denotes the distance of the atoms, $q$ denotes their partial charge, $\varepsilon(r)$ is a distance-dependent dielectric function, $\sigma$ is a constant, and $W$-s are empirically determined weights [1]. The total free energy consists of the intermolecular energy ($i$ denotes a ligand atom, $j$ denotes a receptor atom), the internal energy of the ligand (both $i$ and $j$ denotes a ligand atom), and the internal energy of the receptor (both $i$ and $j$ denotes a receptor atom).

The receptor is often considered rigid, that is, its rotatable bonds are fixed during docking. In this case the distance between receptor atoms cannot change, so the internal energy of the molecule is constant and does not need to be calculated, which simplifies the evaluation of the scoring function.

AutoDock applies a Lamarckian genetic algorithm for optimization, which is a hybrid global-local search method. This means that a new population of possible solutions is generated according to the rules of ordinary genetic algorithms in each generation cycle, then some solutions are subjected to an adaptive local search method similar to hill climbing, which greatly enhances the performance of the genetic algorithm [2].

### *B FPGA suitability*

Due to the random-based search method of AutoDock, dozens of distinct docking runs must be performed for one receptor-ligand complex to obtain reliable results, which may take a few hours even on a high-end PC. On the other hand, the algorithm can be parallelized well, and it is suitable for FPGA implementation. During the search millions of possible binding geometries must be evaluated, which are independent operations. The scoring function must be calculated for hundreds of atom pairs when evaluating a geometry, these operations could be executed simultaneously, as well, which enables to apply pipelines. The algorithm needs a lot of arithmetic operations, but floating-point precision is not required since the chemical model, that is, the scoring function itself is quite inaccurate. All of these facts suggest that an FPGA-based implementation could achieve significant speedup over the original AutoDock software.

## III. Implementation

The algorithm was implemented on an SGI RASC RC-100 blade, which includes a Virtex 4 LX200 FPGA. The implementation has a pipelined structure (Fig. 1). The first stage realizes the genetic algorithm, the second stage calculates the positions of ligand atoms, and the third one consists of two major modules, which are responsible for the intermolecular and ligand internal energy calculation. As a consequence, three different geometries are evaluated in parallel. The four modules require about the same number of clock cycles for one calculation period in case of molecules with typical size and structure, so that they do not have to wait for each other too long.

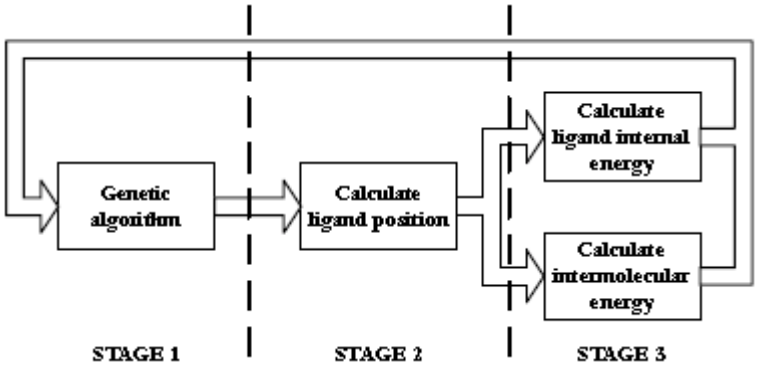



Figure 1: Implementation overview

### *A Genetic algorithm*

The genetic algorithm module maintains a population of solutions in the BRAMs of the FPGA, and generates a new potential solution with selection, crossover and mutation operators, that is, it assigns values to the degrees of freedom of the problem in each calculation period. There are some differences between the implemented and the original algorithm, the most significant one is the applied selection scheme. AutoDock uses proportional selection, which does not fit the FPGA capabilities; therefore a binary tournament selection method is used, which could be implemented

easily in the FPGA. After an entire new population was generated, the module performs the same local search method which is applied in AutoDock.

### *B Ligand position calculation*

The second stage of the pipeline rotates and moves the ligand atoms according to the parameter values generated by the genetic algorithm. The rotation is calculated with quaternion multiplications, which require several normal multiplication and addition operations. However, the majority of the rotations that have to be calculated are independent from each other. In accordance with this, the module consists of a long pipeline, and when it is full, it executes one rotation in each FPGA clock cycle.

### *C Intermolecular energy calculation*

The number of receptor-ligand atom pairs can be very high, which would lead to a time-consuming intermolecular energy calculation. AutoDock applies the so-called grid method [3] to avoid this. Since the scoring function depends only on the types and distances of atoms, if the receptor is rigid, it is possible to determine the energy value for the whole receptor and one ligand atom with known type and position before docking. This value has to be calculated over a 3D grid for each ligand atom type, supposing that the ligand atom is located in the vertices. In addition, another two grids are needed which represent the electrostatic and desolvation terms. During docking, the scoring function does not need to be calculated directly, as the energy contribution of one ligand atom and the receptor can be determined with trilinear interpolation from the pre-calculated grids. As a consequence, the number of required operations becomes proportional to the number of ligand atoms instead of the number of receptor-ligand atom pairs.

In case of the implementation, the grids are stored in four 8 Mbyte SRAMs, which are connected to the FPGA with 64-bit data busses on the RC-100 module. The energy contribution of different ligand atoms can be calculated simultaneously, that is, the required interpolation formula could be implemented with a pipeline in the FPGA, which generates one result in each clock cycle similar to the position calculation module. However, the computation requires several values from the grids for one ligand atom. These SRAM read operations take two or four clock cycles with about the same probability depending on the actual position of the current ligand atom. As a consequence, the number of clock cycles which the module requires for one calculation period is about three times the number of ligand atoms.

### *D Ligand internal energy calculation*

In case of the internal energy calculation, the whole AutoDock scoring function needs to be implemented in the FPGA. The module is pipeline-based, just like the previous ones. However, calculating the internal energy of the ligand requires the scoring function to be evaluated for all non-bonded ligand-ligand atom pairs. This way the number of operations would be proportional to the square of the number of ligand atoms, the module would be much slower than the other ones and it would slow down the whole pipeline in Fig. 1. To avoid this, the module consists of eight identical pipelines, and as a result, it generates and accumulates the energy contribution of eight atom pairs in each clock cycle. As a consequence, the computation time of this module is similar to that of the position and intermolecular energy calculation modules even in case of large ligands.

## IV. Results

The implementation was tested on 58 receptor-ligand complexes, AutoDock was run with the same parameter settings on a 3.2GHz Xeon CPU, and the results were compared to each other. 100 independent runs were performed for every molecule pairs, each of them consisted of 2.500.000 energy evaluations. The results were clustered according to the geometry, that is, similar results were put to the same cluster. The result with the lowest binding energy can be considered as the predicted binding pose, that is, the final result of the docking. The cluster which includes this result

is the best one, its size shows how many times the best pose (or a very similar one) was found out of the 100 run, which indicates the reliability of the predicted geometry. In case of every molecule pairs the experimentally determined binding pose was known, which enables to validate the docking algorithm. A docking is usually considered successful, if the root mean square deviation between the experimentally determined and the predicted binding geometry is below 2 Angströms.

Table 1: Result comparison

| PDB ID | Best energy [kcal/mol] | | RMSD of best result [A] | | Size of best cluster [%] | | Run time of one run [s] | | Speedup |
|---|---|---|---|---|---|---|---|---|---|
| | AD | FPGA | AD | FPGA | AD | FPGA | AD | FPGA | |
| **1ulb** | -6 | -6 | 0,80 | 0,78 | 98 | 99 | 26,3 | 2,01 | 13,08 |
| **1q8t** | -8,01 | -8,08 | 1,91 | 1,88 | 85 | 96 | 55,8 | 2,5 | 22,32 |
| **1cbr** | -9,13 | -9,13 | 1,03 | 1,04 | 96 | 93 | 56,8 | 2,51 | 22,63 |
| **2baj** | -12,53 | -12,58 | 0,56 | 0,52 | 53 | 100 | 75,8 | 3,02 | 25,1 |
| **1hvr** | -15,04 | -15,39 | 0,59 | 0,46 | 8 | 56 | 156 | 4,43 | 35,21 |
| **1u33** | -5,87 | -10,28 | 4,6 | 1,3 | 4 | 20 | 147,7 | 4,2 | 35,17 |

Table 1 shows the results of AutoDock and the FPGA implementation in case of some examples. The two algorithms gave the same or almost the same results in every aspect for the majority of the molecule pairs (*1ulb*, *1q8t*, *1cbr*). In several cases, the energy and RMSD value of the best result were similar, but the size of the best cluster was different; the FPGA implementation found the best geometry much more frequently (*2baj*, *1hvr*). Moreover, the FPGA algorithm was able to identify the true binding pose (RMSD value below 2A) for some receptor-ligand complexes which could not be docked successfully by AutoDock (*1u33*). This means that the overall performance of the FPGA implementation was better than that of AutoDock for this molecule set, which is caused primarily by the differences of the applied genetic algorithm. This does not imply, however, that the performance of the FPGA algorithm would be better for other molecules and with other parameter settings.

The last columns in the table show the run times of one docking run and the speedup of the FPGA over AutoDock. The latter highly depends on the size and structure of the ligand. Usually, the bigger the number of ligand atoms and rotatable bonds, the higher the speedup that can be achieved. For this molecule set, the actual value was always in the range of 10 to 40, the average speedup was 23,3. Currently, this is limited by both the bandwidth of the external SRAMs and the size of the FPGA; higher speedup may be achievable on a more ideal platform.


## Acknowledgement

I would like to give thanks to Dr. Béla Fehér for his useful advices.


## References


[1] R. Huey, G. M. Morris and A. J. Olson, "A semiempirical free energy force field with charge-based desolvation", *Journal of Computational Chemistry,* 28(6):1145-1152, Apr. 2007.

[2] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew and A. J. Olson, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function", *Journal of Computational Chemistry,* 19(14):1639-1662, Nov. 1998.

[3] G. M. Morris, D. S. Goodsell, R. Huey, W. E. Hart, R. S. Halliday, R. K. Belew and A. J. Olson, *AutoDock 3.0.5 User's Guide.*

URL: http://autodock.scripps.edu/

# BLAST ACCELERATION WITH DATABASE PREFILTERING

**Péter LACZKÓ**
**Advisor: Béla FEHÉR**

## I. Introduction

Finding homologies between a genetic sequence of interest and the contents of a bioinformatics database constitutes an important step in many sequence analysis protocols. This task ultimately boils down to finding the most likely alignments of the query sequence to each sequence in the database. While an exact solution to this problem is known [1], in practice a computationally more feasible approach is usually taken, due to the massive and ever-increasing volume of genetic or proteomic information being used as reference to the alignment. This heuristic algorithm, known as BLAST [2], is widely used in current research, since it allows for finding optimal alignments in reasonable time.

However, with a very large reference database and a large amount of query sequences to be aligned, even this fast algorithm can take an impractical time to execute. Various approaches exist to tackle this problem of computational complexity, one of which being the reduction of the reference database size by prefiltering, which in turn reduces the running time of the BLAST algorithm (since it is roughly proportional to the size of the reference database). This paper describes one such prefiltering system, covering both the prefiltering algorithm and its efficient implementation on an FPGA accelerator card.

## II. Prefiltering Algorithm

The first two steps of the BLAST algorithm is to scan through the reference sequences and to find short subsequences ("words") that have a high score when matched to any word of the query, given a predefined scoring scheme. In subsequent steps these high scoring pairs are extended by BLAST to yield stretches of high similarity, called Maximal Segment Pairs, of which the ones that are the most unlikely to have arisen by chance are reported to the user [2].

The first two steps therefore limit the scope of alignment efforts to the proximity of those words that have high similarity to the words of the query in a simple, exact-matching sense. The basic idea of BLAST prefiltering is to reduce the reference database to roughly these regions with a very fast filtering algorithm, so that the actual BLAST run will be run against a much smaller reference database. Since the first steps of BLAST themselves perform a similar task, the advantage of the prefiltering approach lies in the fast and possibly parallelized implementation of the filtering, which should scale better with the reference length than the first steps of BLAST.

The task therefore is to scan through the entire reference sequence and compare each word to each word of the query. The naïve approach would take O(N*M) steps, where N and M are the word counts of the reference and the query, respectively, which are roughly equal to their length, assuming that the word size is a small constant. A much better solution is to create a random access lookup table that is capable of indicating the presence or the absence of all the possible query words. This way, when reading through the reference sequence, each encountered word can quickly be checked against the lookup table, resulting in an algorithm that finishes in O(N+M) time, where M corresponds to the initialization of the table based on the query words, and the N term is due to the necessity to examine the presence or absence of each reference word. Upon a match, a score function is incremented at every position covered by the matching word, and regions with a cumulative score higher than a given threshold will serve as seeds for the filtered input database for BLAST. The process is illustrated in Figure 1.

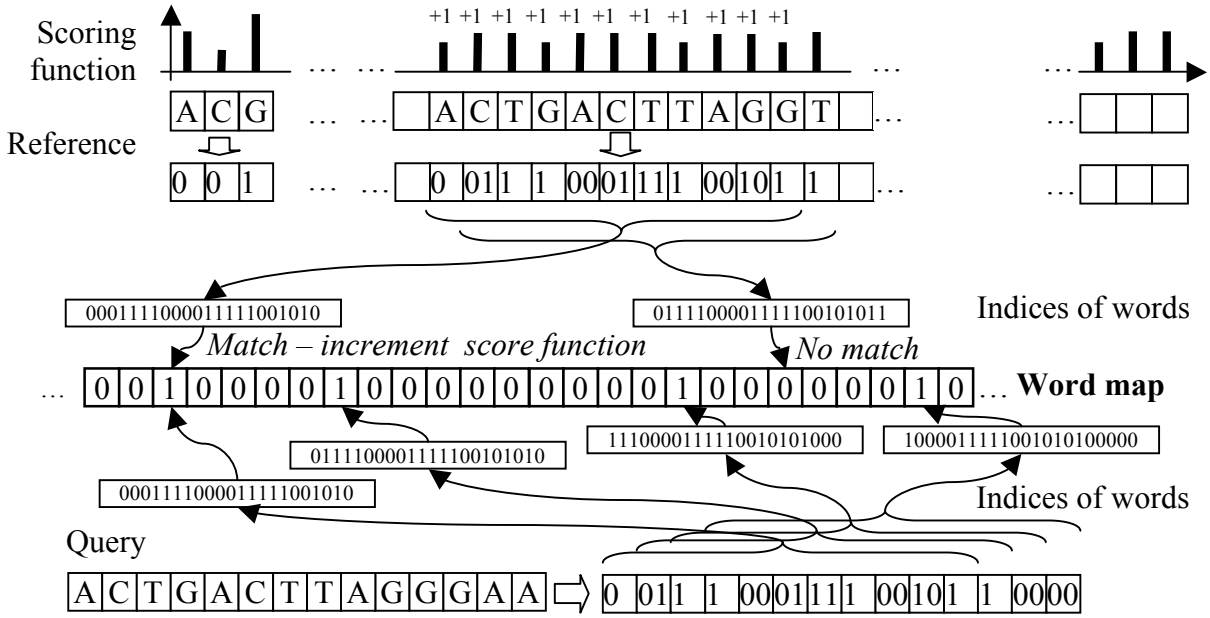



Figure 1: Schema of the prefiltering algorithm

## III. Parallel FPGA Implementation

The prefiltering algorithm must execute as fast as possible, otherwise the performance gain of its use over a conventional BLAST run diminishes. An FPGA accelerator card may be used to exploit the possibilities of parallelization present in the algorithm. One such system with a slightly different implementation is described in [3].

The one-to-one mapping between words and indices (as opposed to e.g. hash tables), given the limited amount of memory available, imposes a natural limit to the maximum word length that can be used for a given type of sequence (nucleotide or protein). However, an algorithm with one bit of information processed per step lends itself to bit-level parallelization. If the individual bits of memory words are filled with the word maps corresponding to different queries, the database may be prefiltered for these queries at the same time, with a single read of the reference. Our system uses 4 blocks of 8 MB SRAM memory, organized into 64 bit words, therefore allowing the concurrent processing of 64 queries. The reference sequence is streamed in letter-by-letter, and the system keeps track of 64 individual scoring functions for the actual word-size region. As a position exits from this sliding window, its score value is compared to a predefined threshold. If above, a one is emitted for the given query, otherwise the output bit is zero. Since 64 queries are simultaneously processed, for every reference position 64 bits are produced as output.

The host side program is responsible for streaming in the sequence data, and for decoding the 64-bit output words corresponding to each sequence position. Furthermore, high scoring regions, represented by stretches of ones in one of the 64 output streams, must be extended to each direction, in order to allow BLAST to extend the likely hit in its third step. The resulting extended regions constitute the filtered database, which is provided as input for the BLAST program.

## References


[1] T. F. Smith and M. S. Watermann, “Identification of common molecular subsequence,” Journal of Molecular Biology, 147: 196–197, 1981.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, “Basic Local Alignment Search Tool,” *Journal of Molecular Biology*, 215( 3): 403–410, Oct. 1990.

[3] P. Krishnamurthy, J. Buhler, R. Chamberlain, M. Franklin, K. Gyang, and J. Lancaster, “Biosequence Similarity Search on the Mercury System,” in *Proc. of the IEEE 15th International Conference on Application-specific Systems, Architectures and Processors*, September 2004, pp. 365-375.

# HYBRID MODELING FOR AN INTELLIGENT GREENHOUSE

**Péter EREDICS**
**Advisor: Tadeusz P. DOBROWIECKI**

## I. Introduction

Greenhouses are structures widely used in vegetable production and for growing flowers. Solar radiation passing through the walls and roofs supplements the heating system in the cold season. In hot weather other actuators, like roof vents, shading or evaporative cooling are used to avoid overheating.

Control systems for greenhouses have not changed much in the last years: each actuator is individually controlled based on set-points. On the other hand the concept of intelligent control evolves around the replacement of set-points by the goals. The needs of the plants can be precisely stated with the required temperature, humidity and radiation patterns. The traditional reactive control can be thus replaced by predictive models [1], where the model of the whole system should be used to predict the effect of the control decisions. The required synchronization of the actuator actions comes naturally with the evaluation of predictive models. The level of synchronization and the performance of the system can be raised even further by the usage of planning.

The necessary basis for the intelligent control is the precise modeling of the greenhouse. Accurate modeling requires large amount of data to be collected from strategically selected locations. Luckily quasi-homogenous thermal zones can be localized in the greenhouse, decreasing the number of the required measurements: Zone-0 is the heating system; Zone-1 is composed of the covered desks; Zone-2 is the air layer below the shading screen; Zone-3 is the air layer above the shading screen and Zone-4 is the external weather. Temperature measurements from a 100 $m^2$ greenhouse are available from all zones for experimental purposes. In Zone-2 and Zone-4 radiation measurements are also recorded. Temperature and cloud coverage records are obtained from an online weather forecast.

## II. Modeling the greenhouse

The goal of thermal modeling is to predict the future temperature values for all important measurement locations. The inputs to the model are the temperature measurements and the states of all actuators. Although in some cases control based on involved analytic parametric models is possible, we aimed at practical situation, where the greenhouse infrastructure is heavily constrained by economical reasons, the character of the production could fluctuate on a weekly basis, and where there is a pronounced lack of expertise in control solutions. In such cases it seems feasible to found the control on the adaptive black-box models. The aim of such model is to predict the thermal state of the desks and the internal zones. Prediction of the external weather is also necessary as it helps in following weather trends in the internal predictions. It seems however reasonable to decompose the models according to the causality relation and to separate the external weather following model. The granularity of decomposition could be raised even further (see Fig. 1), as the resulting modeling problems are easier to handle.

Module-1 in Fig. 1 is responsible for the prediction of the locally recorded external weather. Online weather forecasts have the advantage of serving directly full forecasts along with current measurements. Unfortunately their precision for the actual location of the greenhouse is not acceptable, prediction of trends is however very reliable this way. For a shorter horizon time-series mining can be used on earlier measurements to produce reliable predictions for a few hours ahead. During normal operation Module-2 is deactivated as the predictions come from the online forecast source. However in case of network outage the data has to be restored. The model of the internal state of the greenhouse is

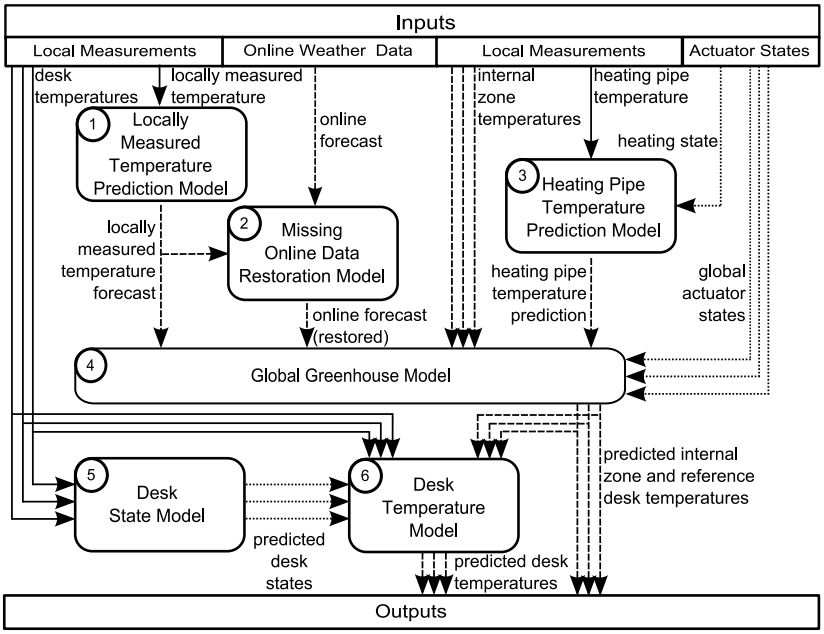



Figure 1: The proposed model decomposition into 6 modules

composed of 4 modules. Module-3 and Module-4 are responsible for all internal zones, but not for the desks: the desks are represented here by a single reference desk. It is used later to calculate the complete state of Zone-1. Application of a reference desk simplifies the global house model, and makes the design flexible to handle less equipped greenhouses. Module-3 predicts the heating pipe temperature, based on the current internal temperature, heat pipe temperature and the heating control signal. This model can be easily separated, as the pipe temperature is mainly determined by its control signal. Module-4 can be realized by a neural network with inputs from the state of the internal zones, the external predictions, the heating pipe temperature predictions and all control signals except heating. It can happen that the desks in the experimental greenhouse can be covered for the protection of sensitive plants. Unfortunately the state of the desks is not recorded. This calls for the application of Module-5 to predict the state of every single desk. Module-6 is responsible for predicting the temperature of the desks based on the current measurements, the approximated state of the desks, the predictions for the reference desk and the prediction for Zone-2.

## III. Results and Conclusion

The hybrid greenhouse model based on the decomposition outlined in this paper is currently under development, and will be tested on the recorded data from the real greenhouse. Module-1 and Module-3 have been already implemented, while implementation of the other 4 modules still requires work based on the measurements and the outputs of the existing modules.


## Acknowledgement

The authors gratefully acknowledge the support of the OTKA, Grant #73496.


## References


[1] Blasco, X., Martineza, M., Herreroa, J.M., Ramosa, C., Sanchisa, J.: Model-based predictive control of greenhouse climate for reducing energy and water consumption. In: Computers and Electronics in Agriculture, pp. 49–70, Amsterdam, The Netherlands (2007)

# NEURAL NETWORK BASED MOBILE ROBOT NAVIGATION

**István ENGEDY**
**Advisor: Gábor HORVÁTH**

## I. Introduction

The navigation system of a mobile robot must handle numerous well separable subtasks for proper operation. One of these tasks is motion planning, which is divided into at least two separate parts, the path planning and the movement itself. The first is the procedure, which designs the path from the current location to the end point, of which the robot will have to go through. The movement is the procedure of keeping the robot on this path. The motion could be complex [1], because the movement of the robot must meet various physical constraints, like the momentum of the car.

The motion planning could be carried out in other ways too. Using soft computing methods, the path planning and the movement can be carried out simultaneously [2]. In this paper we will present an artificial neural network based robot navigation solution, which could avoid moving obstacles.

## II. Navigation method

In our motion planning solution the mobile robot is controlled by an artificial neural network (ANN). It is trained with the backpropagation through time method (BPTT) [3], which is a well known training algorithm of dynamic feedback ANNs. Its use for robot navigation has been already shown by D. Nguyen and B. Widrow [4]. The main idea behind this is to open the feedback control loop and unfold it through many iteration steps, thus making a simple feed-forward system, which can be trained with the usual backpropagation algorithm (Fig. 1. a).

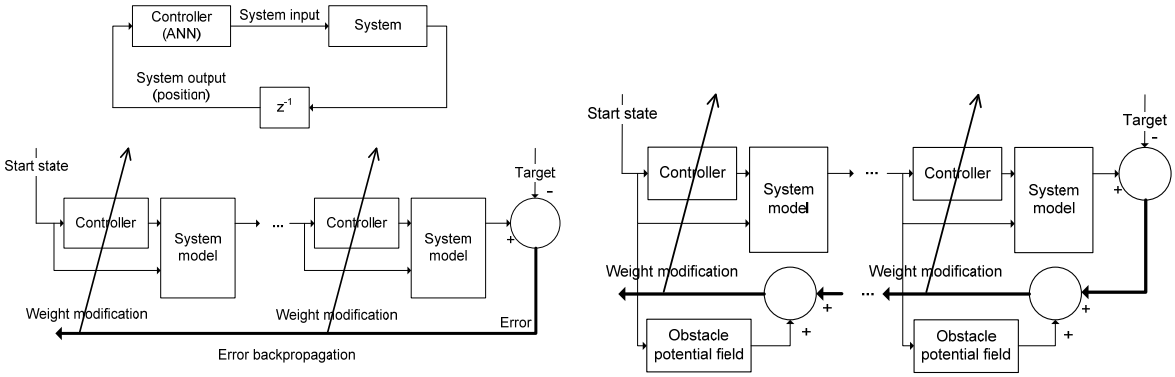



Figure 1: (a.) BPTT in use of training in a control loop (b.) Regularizing the BPTT

This method is able to navigate a mobile robot from any starting point on the working area to any target point. The path and the motion planning are done simultaneously, as the constraints of the robot are taken into account during the backpropagation through the system model. This way the controller is trained to follow a path, with the robot controlling commands calculated. This method however is not able to avoid obstacles, especially not moving obstacles.

## III. Obstacles

To make the BPTT method able to handle obstacles, we have elaborated the following solution. Based on the location and the size of the obstacles, a potential function can be defined (Eq. (1) left), which is used to repel the robot away from the obstacles. This function must be used to extend the

cost function of the Delta-rule. The cost function is regularized with the potential field (Eq. (1) right and Fig. 1. b), so the goal of the weight modification is not only to minimize the error at the end of the simulation chain, but also to minimize the potential of the path, to get the robot the farthest from the obstacles. This way the obstacles could be avoided. The potential field and the regularization of the cost function are defined as follows:

$$U_i(y)=\begin{cases}\dfrac{1}{(d_i(y)-r_i)^2}, d_i(y)>r_i+\varepsilon \\ \dfrac{1+r_i-d_i(y)}{\varepsilon^2}, otherwise\end{cases} \qquad C_R(t,y)=\|t-y_n\|^2+\lambda\sum_{i=0}^{n}\sum_{j=0}^{k}U_j(y_i) \qquad (1)$$

where $U_i(y)$ is the potential field of the $i^{th}$ obstacle, $d_i(y)$ is the distance from the centre of the obstacle, $r_i$ is the radius of the obstacle, $y$ is the position of the robot, and $\varepsilon$ is a small positive constant. $C_R(t,y)$ is the regularized cost function, $t$ is the position of the target, $n$ is the number of iteration steps during BPTT, and $k$ is the number of obstacles.

To use this method to navigate among moving or previously unknown obstacles, further modifications must be made. Till this point, we did not specify, whether the ANN is to be trained offline or online. Offline training could be used, if the obstacles were previously known and static. In such cases the training could be carried out from multiple starting points. In case of moving, or previously unknown obstacles, the use of online training seems to be the only option, despite its trivial drawback: the increased need of computational power.

Online training brings the ability to adapt to changing environment, e.g. avoid moving or to previously unknown obstacles [5]. It has also many advantages. There is no need for training the ANN from multiple starting points, only the current location of the robot should be used, which makes the training much faster. Using the online training makes the method to an anytime algorithm between reasonable limits, because the navigation result is degraded only in quality with the decreasing time limit until the ANN training becomes insufficient. On the other hand this makes the algorithm well-scalable; using a faster CPU can increase the quality of the result of the algorithm.

## IV. Conclusion

In this paper we have shown how the classical BPTT training approach can be extended using regularization, to take additional constraint into consideration, like obstacles. Simulations and real robot experiments have proved that using real-time online training, this method is able to handle moving obstacles as well.


## Acknowledgement

The authors gratefully acknowledge the support of the Hungarian Fund for Scientific Research (OTKA), Grant #73496.


## References


[1] J.-C. Latombe, *Robot Motion Planning,* Kluwer, 1991.

[2] Pratihar D.K., Algorithmic and soft computing approaches to robot motion planning, *Machine Intelligence and Robotic Control*, 5(1):1-16, 2003

[3] M. Minsky and S. Papert. *Perceptrons, An Introduction to Computational Geometry,* MIT Press, 1969

[4] D. Nguyen and B. Widrow, "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks," in *Proc. of the International Joint Conference on Neural Networks*, pp. 2:357-362, 1989.

[5] I. Engedy and G. Horváth, "Artificial Neural Network based Mobile Robot Navigation," in *Proc. of the IEEE International Symposium on Intelligent Signal Processing*, pp. 241-246 ,Budapest, Hungary, Aug. 2009

# ASYNCHRONOUS SAMPLE RATE CONVERTER FOR CLASS-D DIGITAL AUDIO AMPLIFIER

**György DANCSI**
**Advisor: Béla FEHÉR**

## I. Introduction

In a digital audio device sample rate conversion is needed when different sample time signals are processed. The ratio of sample time of the signals can be unconstrained, for example sampling is timed by different oscillators. In this case asynchronous sample rate converter (ASRC) can be used. To be able to read the underlying continuous signal between the samples is required, just like to create sub-sample length delay. This method also provides to synchronize the input samples to local clock.

When restoring the analog signal from the sequence of samples, the accuracy of the time points at which the samples are converted is as important as the accuracy in the amplitude domain. Typically the connection from a digital audio source to an external digital-to-analog converter is done with a single coaxial wire or optical fiber, carrying the coded serial bitstream according to the S/PDIF standard. In the digital-to-analog converter, this signal is processed by some receiver chip, which has as main task to regenerate a clock signal from the data stream with PLL. This is the point where the "jitter" is created.

A high performance digital-to-analog converter or digital audio amplifier requires an ASRC to synchronize the input digital data stream to low jitter system clock. It is highly desirable to able to integrate the ASRC function with the other signal processing functions of the digital amplifier [1].

In this paper I focus the FPGA realization of a sample rate converter intellectual property (IP) using polynomial interpolation method for upsampling. I would like to use this IP to produce constant sample time oversampled audio from every standard source in a custom digital PWM modulator, part as a digital input Class-D amplifier which presented in [2]. In that essay new results in uniformly sampled PWM correction and direct digital feedback of the audio output signal are introduced. Noise shaping and resampling based predistortion have been applied. The closed loop control have been implemented in 2-DOF (two degree of freedom) structure.

## II. Asynchronous Arbitrary Sample Rate Conversion

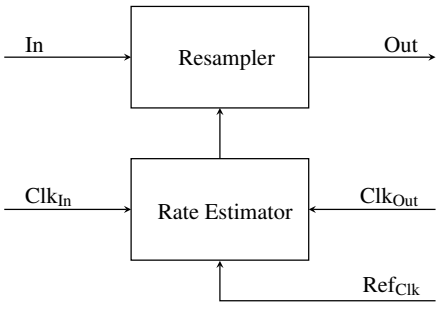



Figure 1: ASRC architecture

In the ideal case, if the input is time continuous signal $x(t)$ sampled at $f_{si}$, the output will be identical to $x(t)$ sampled at a different rate $f_{so}$. If the sampling rates are derived from different clocks, the conversion ratio will be arbitrary or even slowly time-varying.

### A. *The concept of sample rate conversion*

The source samples are interpolated to a heavily oversampled intermediate signal using a digital filter. To allow a conversion between arbitrary sample rates this intermediate signal is transform into a pseudo-analog representation by holding the value in-between the interpolated samples. The resulting continuous-time signal is then resampled at the sink sample rate.

To understand how to produce the pseudo-analog signal with fractional delay using a discrete-time

system it is necessary to discuss interpolation techniques. Interpolation of discrete-time signal is based on the fact the amplitude of corresponding continuous-time bandlimited signal changes smoothly between the sampling instants.

*B. The arbitrary sample rate converter architecture*

Practically the problem will be partitioned into a freqency tracking unit, which determines precisely the sink phase relative to the source samples, and a digital interpolation filter. This shown in the Fig. 1. A buffer is used as input sample storage. For frequency tracking a fast counter can be used, which runs independetly of the source and sink sampling clock. Average can be applied on the measured period times. The relate of the two values determined by the rate estimator. The accumlated ratio means the inverse of the actual phase, or more precisely the fractional delay parameter, which is denoted $-D$ in (3) (see in III. A.).

## III. Used Methods

*A. Candan's structure for Lagrange interpolation*

Lagrange interpolation is based on determinating the $N^{\text{th}}$-order polynomial passing through $N+1$ sample points. Zero-order hold, linear and cubic interpolation are some special cases of Lagrange interpolation. In the case of an *infinite* number of *equally spaced* samples, the Lagrangian basis polynomials converge to shifts of the sinc function (Shannon reconstruction formula). Lagrange fractional-delay filters are maximally flat in the frequency domain at DC.

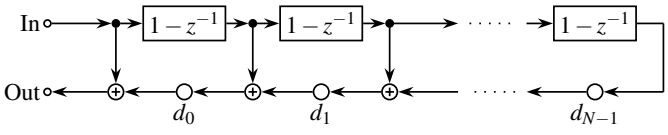


Figure 2: The proposed structure

$$\left.\frac{\mathrm{d}^m\left(H^*(e^{j\omega})-H(e^{j\omega})\right)}{\mathrm{d}\omega^m}\right|_{\omega=0}=0 \quad \text{for } m=0,1,\ldots,N\,, \tag{1}$$

where $\omega$ is the angular frequency, $H(e^{j\omega})$ is the transfer function of the filter and $^*$ notes the complex conjugate.

Farrow has proposed a filter bank structure with intermittent delay multipliers [3]. Candan's structure based on discrete time Taylor series expansion [4]. Let $\Delta$, the backward difference operator: $\Delta f[i] = f[i]-f[i-1]$, is dual of derivative operator. The discrete time dual of polynomial powers are defined as fallows: $x^{\underline{N}} = x(x+1)(x+2)\cdots(x+N-1)$. So the discrete time dual of Taylor series can be written

$$f(t)=\sum_{n=0}^{\infty}\Delta^n f[i]\frac{(t-i)^n}{n!}. \tag{2}$$

The consecutive terms in summation can be recursively calculated as follows (Horner scheme)

$$\frac{(-D)^N\Delta^N}{N!}x[i]=\frac{(-D)^{N-1}\Delta^{N-1}}{(N-1)!}\,\frac{(-D+N-1)\Delta}{N}x[i]. \tag{3}$$

When the recursion is inserted in the Taylor summation, the overall structure simplifies to the structure shown in Fig. 2. The gain of the multipliers depend on the $-D$ parameter:

$$d_n=\frac{-D+n-1}{n+1} \quad \text{for } n=0,1,\ldots,N-1\,.$$

The computational complexity of the structure is $3N$ additions and $2N$ multiplications for the $N^{\text{th}}$ order interpolation. A novel feature of the structure is the run time increment–decrement possibility of the interpolation order.

*B. Optimal design based on genetic algorithm*

A polynomial interpolator can be thought of as a filter with continuous-time impulse response. From the impulse respose the frequency response can be calculated. The quality of the interpolation as the signal-to-noise ratio (SNR) determined by the frequency response. Using the SNR or the the weighted SNR as quality measure, it is possible to design an optimal interpolator of chosen oversampling ratio and order [5]. This method provides better SNR at lower oversampling ratio according to other polynomial interpolations (See Fig. 3). The offered coefficients has no symmetric property like the Lagrange fractional delay filter. Additionally these filters are not maximally flat at $\omega = 0$, and the interpolated curve will not necessary go through the points.

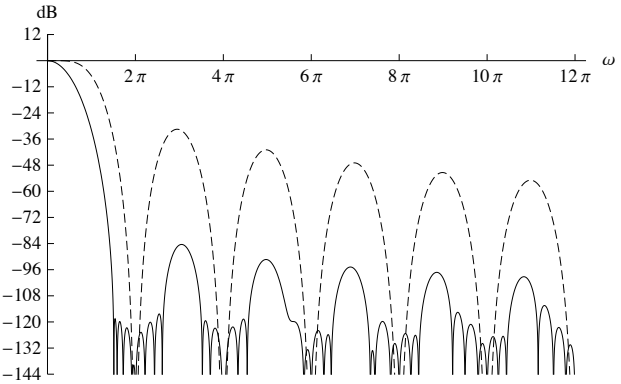



Figure 3: The frequency response of the 5$^{th}$ order interpolators: Lagrange (dashed line) and optimal (continuous line)

## IV. FPGA Implementation

Xilinx has an ASRC reference design [6]. It can be used for up and downconvert too. It uses a prototype filter, and calculates the correct phase of the filter and do the FIR interpolation in real-time. It has been written in Verilog hardware description language. I remade this design in the System Generator for DSP graphical block diagramming tool, which is a MATLAB Simulink block library. The sizing has been made only for upsampling operation. The output sample clock is related to the processing clock (two's power), so some components might have been abandoned. Fig. 4 shows the main blocks. The model can be exported from System Generator in NGC netlist form.

The input samples stored in a dual port Block RAM, which is used as a FIFO. The difference of the write and the read address means the FIFO level and there is a desired FIFO level. The write address is incremented when a new input sample arrived. The read address is defined by a counter.

*A. Ratio detection and control*

The period of input clock is measured over 1024 cycle, and a moving average filter is applied on the measured value. This value is divided by the output period (simple shift operation) to obtain a calculated ratio. The inverse of the ratio is also needed, it is calculated by a multi-cycle pipelined divider. In the reference design the ratio was the controlled variable. It was replaced to the inverse ratio, so the slow division has taken out from the closed loop control. The process has integral property, so there is no need to add integral term to the controller in order to eliminate the residual steady-state error.

The ratio control uses one of two algorithms depending on whether the input rate is changing. The rate-change tracking mode is used to quick interception, and to adjust the level of the input FIFO to the desired level. In locked mode the amount and the rate of change of the ratio is limited. This mode can track small drift in the input clock frequencies.

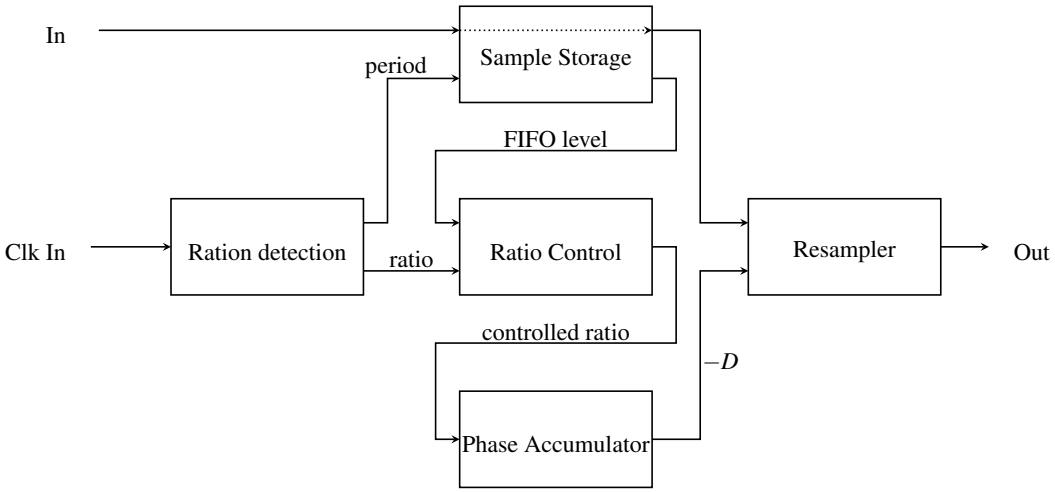



Figure 4: The main build blocks of the ASRC

### *B. Resampler*

The resampler creates a set of samples from the input samples based on the output-to-input ratio produced by the ratio detection and control unit.

When a new input sample arrived into resampler the differentiators calculate the differences (one per clock cycle). When every differences are known the actual output can be calculated according to (3).

## V. Results

The presented asynchronous sample rate converter can be integrated into the Class-D amplifier. There is no need for external receiver circuit and clock multiplier PLL any longer. The constant output sample time makes the controller more robust again the parameter change of the plant.

The working of resampler can be simulated easily in Simulink. Primary I tested the performance of the ratio control, the output spectra isn't measured. S/PDIF receiver IP has been used as input source. The interception happened in a flash. When the oscillator has been heated, the measured frequency is slowly changed, while the ratio control unit was providing continually ratio tracking. The choice between the implemented resampler units will be made after listening.

## Acknowledgement

The author is grateful to Gyula István Nagy — the co-author of [2] — for useful ideas and discussions.

## References

[1] P. Midya, B. Roeckner, and T. Schooler, "Asynchronous sample rate converter for digital audio amplifiers," in *121$^{th}$ AES Convention*, 2006.

[2] Gy. Dancsi and Gy. I. Nagy, "FPGA alapú D-oszályú erősítő direkt digitális szabályozással [FPGA Based Class-D Amplifier with Direct Digital Feedback]," in Hungarian, Scientific Student Conference Paper, 2007.

[3] C. W. Farrow, "A continuously variable digital delay element," in *Circuits and Systems, 1988., IEEE International Symposium on*, pp. 2641–2645 vol.3, 1988.

[4] Ç. Candan, "An efficient filtering structure for lagrange interpolation," *IEEE Signal Processing Letters*, 14(1):17–19, Jan. 2007.

[5] O. Niemitalo, "Polynomial interpolators for high-quality resampling of oversampled audio," 2001.

[6] Xilinx Inc., *XAPP514 – Virtex-II Pro/Virtex-4 Audio/Video Connectivity*, Oct. 2008.

# VARIABLE PRUNING IN BAYESIAN SEQUENTIAL STUDY DESIGN

**Gergely HAJÓS**
**Advisors: Péter ANTAL, Tadeusz DOBROWIECKI**

## I. Introduction

In this paper we present a full bayesian method to take in account costs and utility of data used in induction. First we introduce the stopping problem, then the multi-armed bandit problem, then the Markov decision process. After the theoretical foundations we briefly discuss the two research areas connected to these problems: the adaptive study design, and the active learning. Finally we present the extension of this adaptive study design in a multivariate context for the problem of feature subset selection (FSS). The application area is the sequential study design of partial genome screening studies (PGAS).

The *optimal stopping problem* is defined by a sequence of independent random variables $X_1, X_2, \ldots, X_i$ with known distribution and a score function $f(\cdot)$ which gives a score to the observed sequence of random variables ($y_i = f(X_1, X_2, \ldots, X_i)$). In each step $i$ a decision is made whether to stop and get score $y_i$ or continue observing. The objective is to maximize expected reward. A special case of the optimal stopping problem is the *classical secretary problem*, which is defined as follows: there is a secreterial position to fill; there are $n$ applicants; at each interview the applicant is accepted or rejected; the rejected applicants can not be accepted later; only one person can be accepted; the applicants can be ranked unambiguously; if the best applicant is chosen the reward is one otherwise zero.

If we can control the type of observations beside sample size, then we can formulate the *multi-armed bandit problem* (K-bandit problem). It is defined by $K$ independent random variables $X_1, X_2, \ldots, X_K$ with unknown distribution. In each step $i$ one has to select and sample one of the random variables $x_i$; the objective is to maximize the sum of samples $y = \sum_{i=0}^{M} x_i$ for a fixed $M$.

The multi-armed bandit problem is formally equivalent to the one state Markov decision process. The general *Markov decision process* is defined by: set of states $S$; set of actions $A$; conditional probabilities $Pr(s_{t+1} = s' | s_t = s, a_t = a)$ which in state $s$ gives the probability of next state $s'$ selecting action $a$; and utility function $U(s_{t+1} = s', s_t = s)$ which gives a reward for state change from $s$ to $s'$. If a finite $M$ step is assumed a possible objective is to define a $d(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ decision function to maximize $y = \sum_{i=0}^{M} Pr(s'_i | s_i, a_i) U(s'_i, s_i)$, where $s'_i = d(s_i, a_i)$.

The modern large-scale formalization of this line of research led to the concept of active learning and adaptive (sequential) study design. In this paper we follow the terminology of sequential study design (SSD).

The objective of the *sequential study design* is to retrieve statistical information from data collected sequentially, given a cost, utility and a budget constraint for data collection. A possible application is for instance to check the effectiveness of drug treatment or find association between genetic factors and diseases, etc. On the field of sequential study design generally the standard hypothesis testing approach is used [1]. The aim is to collect the minimum amount of data necessary to make a decision between null and alternative hypothesis, since one wants to minimize the costs of measurements performed in the study design. In every step of the sequential study design the tests of hypothesises are performed on a set of samples, if one of the hypothesises are accepted the study design is stopped and a decision is made. If a decision can not be made due to lack of enough information the study design continues and more data is collected.

Instead of applying the standard statistical approach in this paper we present a multivariate extension

based on Bayesian networks. In every step of the study design we first approximate the utility of computed results based on available data, second we predict future data based on available data using Bayesian model averaging over Bayesian networks. With the help of future data we predict the utility of the continuation of the study design. If the predicted utility of continuation is bigger than the utility based on available data then the sequential study design is continued, otherwise stopped and the last computed results are reported.

Beside selecting the sample size to minimize cost, in this paper we present an *active learning* approach to reduce the number of variables in every step of the sequential study design [2]. Since the algorithm in every step narrows down the set of investigated variables, in the subsequent step just the last (narrowest) set is used for further analysis. In this way in every subsequent step less and less measurements are performed.

In the case of partial genome association studies (PGAS) contrary to genome-wide association studies (GWAS) only partial information is available about the genome of the participants. In PGAS, we attempt to discover from subsequent measurements of well-selected blocks of variables the relevant genetic factors for a given target set with interim analyzis and meta-analyzis of the available aggregated data sets in order to interpret and guide further measurements (see Fig. 1). The phases are shown in Fig. 1, starting with the GWAS layer and the application of gene prioritization systems for the subjective, knowledge-rich initiation of our pruning process [3].

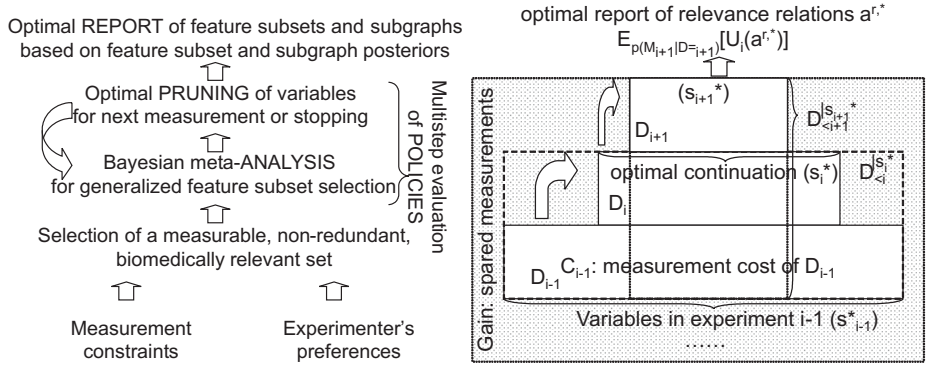



Figure 1: Left: The phases of sequential study design. Right: The main steps of Bayesian pruning in adaptive study design (for notation see Section II.)

In our approach we target the feature subset selection (FSS) problem in the Bayesian context. We apply complex Bayesian network based structural features in the analysis, specifically Markov Blanket Memberships (MBM), Markov Blanket Sets (MBS), and Markov Blanket Graphs (MBGs) [4]. The applied Bayesian multilevel relevance analysis means a multivariate approach to discover relevant sets of variables together with their interactions (conditional and contextual relevance), in correspondence we work with multivariate preferences (utilities). We evaluate typical policies in association studies based on interim Bayesian meta-analysis, and also the performance of a one-step look ahead approximation of the expected value of experiments in the full Bayesian approach. Our application domain is the investigation of genetic background of asthma using PGAS, where the sample collection and genotyping costs are considerable, and a multivariate approach is essential due to complex, weak interactions behind multifactorial diseases.

## II. BAYESIAN SEQUENTIAL STUDY DESIGN AND VARIABLE PRUNING

Since in case of variable pruning beside the selection of the number of samples, we also narrow down the number of variables, we assume the following: a prior $p(M)$ for generative models;variable set $S_i$, where $S_i$ contains only variables present in step $i$ due to the reduction of variables; a corresponding

likelihood $p(D_i|M)$ for the $i$th step and data set $D$ where $D_i$ represents the data set narrowed down to variable set $S_i$ with the samples $N_i$ collected in step $i$; a set of actions, continuing sequential study design consisting of both the selection of $S_{i+1}, N_{i+1}$ or reporting actions (stop experiment and report the last computed model); a context-free, e.g. timeless, cost $C_{N_i}^{S_i}$ of measuring (observing) $D_i$ and a utility function $U(M_0, M^*)$ for stopping and reporting $M^*$ in case of original model $M_0$. While $D_{<i}$ represents the data set narrowed down to variable set $S_i$ with all the samples collected in steps $< i$ ($N_{<i} = \sum_{j=0}^{i} N_i$).

In the optimal Bayesian approach, at step $1 < i$ one possibility is to stop and report the optimal maximal utility model $M^*$ with utility

$$U_i^{\text{report}} = E_{p(M|D_{<i})}[U(M, M^*)] - \sum_{j=1}^{i-1} C_{N_j}^{S_j}. \tag{1}$$

The other option is to continue by selecting the next, optimal experiment defined by selection of $N_i$, with utility

$$U_i^{\text{cont}} = U(N_i) - \sum_{j=1}^{i-1} C_{N_j}^{S_j}, \tag{2}$$

where $U(N_i)$ denotes the expected utility of experiment. In case of a decision problem with finite horizon, backward induction can be applied to calculate Eq. 2. The exponential number of potential future subsequent data makes however the estimation of these expectations computationally prohibitive (for nonmyopic evaluation of the value-of-information, see [5, 6] ). The one-step, myopic approximation of $U(N_i)$ is as follows

$$U(N_i) \approx E_{p(D_i|D_{<i})}[E_{p(M|D_{\leq i})}[U(M, M^*)]], \tag{3}$$

which means that after the first step, the optimal Bayesian decision, (reporting $M^*$ or continuing with measuring $N_i$) can be determined by comparing $U_i^{\text{report}}$ to $U_i^{\text{cont}}$. For an overview and an upper bound for the expected value of an experiment, see [7]. Note that the framework of Markov decision processes is not directly applicable to this context, because of the dynamic state space.

## III. MULTILEVEL ANALYSIS

We assume that there is a special set of target variables, and the goal is the identification of relevant variables, optimal sets of relevant variables, and their interactions (for an overview of FSS problem, see e.g. [8]). The goal of the analogous Bayesian FSS can be defined as the computation of the posteriors for the pairwise relations, relevant sets, and interactions, which can be formalized as the posteriors for MBM, MBS, MBG [4].

Respectively, we assume that the earlier utility function for the model can be decomposed into three parts, specifically for the MBM, MBS, and MBG levels. Note that given the utility function and the posterior over the feature space in step $i$, the expected utility of reporting a structural feature $\hat{f}$ can be computed and the feature value with maximal utility can be determined:

$$f^* = \arg\max_{\hat{f}} E_{p(f|D_{<i})}[U(\hat{f}|f)].$$

## IV. RESULTS

One of the main motivations of the paper is to support more efficient measurements in partial genetic association studies. With this aim we evaluated three policies see Fig. 2. As the sensitivity curves

indicate after the first two pruning steps the external reference variables are still kept, e.g. in case of the greedy method the number of the pruned variables are $66$ and $24$, whereas all the MBS members are included. This means roughly that instead of measuring two times $116$ variables, this allows the identification of the variables measuring only $116 + 66$ variables (i.e. saving one-quarter of the measurements).

For the artificial data set the policies could identify more than 69% of the relevant variables using $< 152$ measurements in three steps, which can be compared to a round robin scheme with $4 * 116$ measurements.

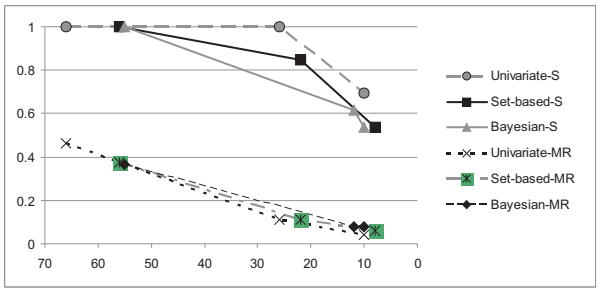



Figure 2: The horizontal axis shows the size of the selected sets in subsequent steps, the vertical axis shows the corresponding misclassification rate and sensitivity for univariate and set-based greedy, and the Bayesian policies in case of the artificial data set.

## V. CONCLUSION

In the paper we investigated the decision support for sequential partial genetic association studies, which are essential methods to ensure high sample size for more targeted statistical analysis after GWAS-based explorations. We demonstrated that the multivariate generalization of the multi-armed bandit problem and budgeted learning — well-known in pharmacology and diagnostics — is viable in the sequential study design context as well, i.e. when both the predictive sampling and the evaluation are multivariate based on Bayesian networks. Preliminary results in an artificial context derived from real-world data indicate that significant saving is possible retaining high sensitivity.

## References


[1] I. R. Konig and A. Ziegler, "Group sequential study designs in genetic-epidemiological case-control studies," *Hum. Hered.*, 56:63–72, 2003.

[2] J. Li, "Prioritize and select snps for association studies with multi-stage designs," *Journal of Computational Biology*, 15(3):241–257, 2008.

[3] S. Aerts, D. Lambrechts, S. Maity, P. V. Loo, B. Coessens, F. D. Smet, L. Tranchevent, B. D. Moor, P. Marynen, B. Hassan, P. Carmeliet, and Y. Moreau, "Gene prioritization through genomic data fusion," *Nature Biotechnology*, 24:537–544, 2006.

[4] P. Antal, A. Millinghoffer, G. Hullám, C. Szalai, and A. Falus, "A bayesian view of challenges in feature selection: Feature aggregation, multiple targets, redundancy and interaction," *JMLR Proceeding*, 4:74–89, 2008.

[5] H. D., E. Horvitz, and B. Middleton, "An approximate non-myopic computation for value of information," in *Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence (UAI'91)*, pp. 101–107. Morgan Kaufmann, 1991.

[6] W. Liao and Q. Ji, "Efficient non-myopic value-of-information computation for influence diagrams," *International journal of approximate reasoning*, 49:436–450, 2008.

[7] J. M. Bernardo, *Bayesian Theory*, Wiley & Sons, Chichester, 1995.

[8] Y. Saeys, I. Inza, and P. Larranaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, 23(19):2507–2517, 2007.

# CONVERGENCE PROPERTIES OF DIRECTED NETWORKS

**Mihály BÁNYAI**
**Advisors: Fülöp BAZSÓ, György STRAUSZ**

## I. Introduction

Data can often be represented with the aid of directed graphs, therefore graph theoretical notions and techniques and network analysis tools are useful for data mining and description of complex systems. Global graph properties like edge betweenness are usually based on the notion of shortest paths, as these give a functional skeleton of the network. We define a new edge-based measure called convergence degree. This measure is related to information processing properties of the network, and is based on shortest paths, thus have the potential to relate network function and structure.

A common problem with graphs is that we cannot tell if a network is identical to another one just by looking at the points-and-lines figure. That is why we need a representation that is invariant under isomorphic transformations, a so-called network fingerprint. Based on our new measure we introduce a fingerprint that contains some information about the functional role of the network edges and nodes.

## II. Methods

### A. *Definition of convergence degree*

Convergence degree (CD) was introduced in [1] for the analysis of cortical networks and was applied to some random networks [2]. We slightly modified the definition introduced therein, in order to capture more detailed structure of shortest paths. Let $SP(G, e_{i,j})$ denote the set of all shortest paths which contain the chosen edge $e_{i,j}$. So chosen shortest paths uniquely determine two sets, $Out(i, j)$, the set of all nodes from which the chosen shortest paths start, and the set $In(i, j)$ in which the chosen shortest paths terminate. Using these sets we define strict $SIn$ and $SOut$ sets by substracting the intersection of the two sets from both of them. Using these notations we define the convergence degree as follows:

$$CD(i,j) = \frac{|SIn_{i,j}| - |SOut_{i,j}|}{|In_{i,j} \cup Out_{i,j}|} \tag{1}$$

### B. *The node-reduced representation*

We take all the incoming and outgoing edges of a single node and sum up the positive and negative CDs on these edges separately. In the 2D representation the two axes will refer to the incoming and outgoing edges, so we get four coordinate pairs for every node. This representation can serve as a network fingerprint as it maps the nodes of isomorphic networks to the same spatial pattern. To make the representations of networks with different size comparable, we normalize it by the respective in- and out-degrees to fit into the $[-1, 1]^2$ square.

## III. Results

We analysed the large-scale network of the macaque visuo-tactile cortex. In this network nodes represent cortical areas and edges represent anatomical connections. Cortical networks are traditionally said to be small-world networks. However, the notation of small-world networks are very loose, based on heuristic constraints on the average shortest path length and the clustering coefficient. To give a more refined classification, we compared the convergence properties of some random networks and the cortical network. Results are summrized in Table III..

| Graph | clustering coeff. | diameter | avg. shortest path | CD |
|---|---|---|---|---|
| ER | 0.550, $2 \cdot 10^{-3}$ | 3.1, $3 \cdot 10^{-2}$ | 1.88, $3 \cdot 10^{-3}$ | $-3 \cdot 10^{-3}$, 0.23 |
| sw | 0.600, $1 \cdot 10^{-3}$ | 3.06, $2 \cdot 10^{-2}$ | 1.89, $3 \cdot 10^{-3}$ | $2 \cdot 10^{-3}$, 0.54 |
| psw | 0.623, $3 \cdot 10^{-3}$ | 4.32, $8 \cdot 10^{-2}$ | 1.93, $7 \cdot 10^{-3}$ | $1.6 \cdot 10^{-2}$, 0.64 |
| mVT | 0.517 | 5 | 2.15 | $2 \cdot 10^{-2}$, 0.57 |

Table 1: ER denotes Erdős-Rényi, sw denotes small-world, swp denotes small-world with preference, mVT denotes macaque visuo-tactile cortex. For all graphs $|V(G)| = 45$, $|E(G)| = 463$, and reciprocity is set to 0.8.

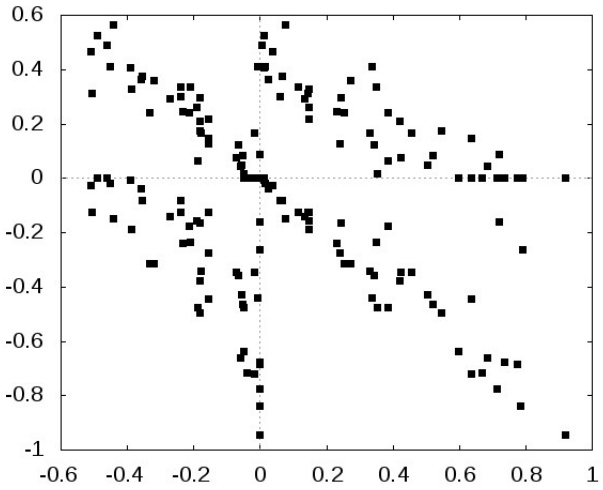



Figure 1: Node-reduced representation of the VT-cortex

We generated an Erdős-Rényi graph, which had edges sampled from a uniform distribution, a small-world graph by the Watts-Strogatz algorithm with the modification that ensures the presence of a Hamiltonian circle and another small-world graph with preferential edge sampling, meaning that a node with high in- or out-degree is more likely to get new incoming and outgong connections respectively. Results clearly show that the preferential small-world algorithm generates the most cortical-like network regarding the global and convergence properties.

We show the node-reduced representation of the cortical network in Figure 1. The hyperbolical pattern in the upper right and lower left quadrant means that if a node is good at integrating information then it's likely to be bad at distributing it. This strucure refers to a hierarchical organization.

## IV. Conclusion

We showed that the convergence degree statistics of a network is a suitable tool for the analysis of the relationship between network function and its structure. For directed networks our approach enables formulation of more accurate models of real networks. The node-reduced representation of the CD can serve as a network fingerprint allowing functional visualisation and clasification of complex networks.


## Acknowledgement

I am grateful to László Négyessy, Tamás Nepusz and László Zalányi for their contribution and insightful comments on our work.


## References


[1] L. Négyessy, T. Nepusz, L. Zalányi, and F. Bazsó, "Convergence and divergence are mostly reciprocated properties of the connections in the network of cortical areas," *Proc. R. Soc. B*, 275:2403–2410, 2008.

[2] M. Bányai, T. Nepusz, L. Négyessy, and F. Bazsó, "Convergence properties of some random networks," in *Proc. of the IEEE International Symposium on Intellingent Systems and Informatics*, pp. 241–245, Subotica, Serbia, 2009.

# PROBABILISTIC MODELING OF UNCERTAINTY IN GENOTYPING STUDIES

**Peter Sárközy**
**Advisors: Péter Antal, Tadeusz Dobrowiecki**

## I. Introduction

In this article we present a probabilistic approach to model uncertainties in genotyping with an explicit representation of rejection and a probabilistic framework to cope with such uncertain data.

While the use of noise models is common in gene expression data analysis, where they cope with similar normalization and feature extraction problems [1], such models are missing in genetic association studies (GAS). Beside measurement problems, uncertainty also arises in haplotype reconstruction as well as later stages of the GAS analysis chain. Our planned framework will allow the explicit propagation of uncertainty from measurement through haplotype reconstruction to data, and allow us to fully utilize all of the information contained in a genotyping measurement.

## II. The measurement

The DNA segments containing the single nucleotide polymorphisms (SNPs) in question from multiple samples are amplified by a polymerase chain reaction (PCR), and then pipetted into spots on a plate that contains their complementary strands bound to the plate substrate. The strands are marked during PCR by different color dyes to denote whether the SNP is wild type or mutant type. The dyes fluoresce under specific monochromatic frequencies, thus the images can be recorded. Each dye emits light of a different wavelength. Some systems use a third control dye mixed in to act as a control value and can be used for normalization. Brightness is nonlinear with the amount of hybridized marked strands.

Each plate is made up of a large number of wells (48-384), which contain the DNA from a single sample, and contain multiple spots (12-64) for each SNP being measured. Different batches of chemical agents are used on a plate, to insure that a single bad chemical doesn't cause complete failure, and also for optimizing the pipetting system's workflow.

## III. Image processing and clustering

In our studies we analyzed two different genotyping systems, Beckman Coulter's SNPstream and the Applied Biosystems' (ABI) TaqMan probe based assay. These systems advertise high call rate and high data concordance, but in practice these results vary. The noise characteristics in the image processing phase of both systems is quite similar, as well as the sample layout and physical characteristics of each measurement. We derive quality metrics such as circularity, noise levels, evenness, signal-to-noise ratio, etc. from each sample based on the image characteristics that we can later use in the quality assessment of each sample.

A 3 level noise model can be applied to the systems, with noise parameters applicable at the plate level (total background noise, intensity offset, chemical batch errors), well level (local noise, large artifacts, dust) and spot level (total intensity, circularity). We assume a multivariate quantized errors-in-variables model for this noise, as seen in Eq. (1).

$$y_t = \log(\underline{x}_t^*) + \underline{\varepsilon}_t,\ \underline{x}_t = \underline{x}_t^* + \underline{\eta}_t. \qquad (1)$$

In the clustering part of the measurement, where we take all of the samples for one SNP and plot the intensities of each dye to produce genotype calls, the two systems use different coordinate

systems, but these are easily transformed. The samples are grouped into clusters to determine the genotype associated with each one. The clusters distribution characteristics and the samples position relative to its corresponding and neighboring clusters lets us derive the final certainty score for the sample. The problem is a classification problem with rejection [2], where we are trying to get the most reliable results out of our dataset, while preserving its integrity. We must not only classify and provide the quantized classification or rejection result, as do most commercial solutions, but also must provide our level of confidence in our decision. Changing the rejection threshold will result in higher area under the curve of the receiver operating characteristic.

## IV. Results of the analysis

We were able to provide more accurate genotype calls than commercial solutions, from the image data alone. We were also able to assign uncertainty data to each sample which correlated very closely with the errors made by other commercial genotyping software. We had marked success on calling genotypes on hard to analyze SNPs.

We compared 768 samples of a single SNP. We used a validated set from a TaqMan based probe (very accurate but slow and expensive) of this SNP. The comparison was between the calls made by the SNPstream application suite and our system The TaqMan probe based system was used as a reference, because its primer is highly optimized for a single SNP and generates very accurate calls, unlike the SNPstream system (48 primers per plate). The SNP chosen for this was one that was difficult to assay with SNPstream, because of its low average spot intensities and high noise levels.

SNPstream called 72 SNPS erroneously out of 768 compared to the TaqMan assay, while our application only called 56 errors.

Altogether there were 36 instances where our application had produced calls different from the SNPstream application, and all of these instances had very high associated uncertainty metrics. The distance from cluster center and the signal-to-noise ratio were the most significant. All the spots that were called differently had very high uncertainty metrics, thus likely to be false.

We found several errors in their measurement protocol. The plates were not washed well enough from residual non-hybridized fluorescent primers, as well as there being wipe marks on the plates. These resulted in many erroneous calls from large noise artifacts. We also managed to pinpoint that the pipetting machine applied gradually increasing concentrations of chemicals on the plates, because the contents of the vials settled fast. Adding a mixing step before each sample load will eliminate this problem.

## V. Further plans

Quality measures can be successfully used to revise previous measurements with a variable certainty threshold, and to create probabilistic models for genotype calls that can aid in haplotype reconstruction. Applying these methods to large sets of data can allow us to refine probabilistic models that can be used for other uncertain data sets as well [3]. Conventional methods of utilizing the output of genotyping calls can be augmented by running these methods multiple times with a variable certainty threshold.

## References


[1] E. Wit, J, McClure, “Statistics for Microarrays”, *Department of Statistics,University of Glasgow, UK*, pp. 57-62, 2004

[2] P. Antal, G. Fannes, D. Timmerman, Y. Moreau, B. De Moor, “Bayesian applications of belief networks and multilayer perceptrons for ovarian tumor classification with rejection” *Artificial Intelligence in Medicine 29*, pp. 39–60, 2003

[3] E. Halperin et al., “Tag SNP selection in genotype data for maximizing SNP prediction accuracy”, *Bioinformatics,* vol. 21, no. 1, 2005

# INFRASTRUCTURE FOR MODEL-BASED CONTROL OF DISTRIBUTED IT SYSTEMS

**Gergely János PALJAK**
**Advisors: András PATARICZA, Tamás KOVÁCSHÁZY**

## I. Introduction

Modern system management applies a feedback control loop scheme (Fig. 1) for guaranteeing a high level of service by (re-)allocating redundant resources in the system to critical functions.

Such feedback control in autonomic computing continuously monitors the service level, for instance performance and availability, and upon an unacceptable deviance triggers health maintenance reconfiguration actions according to a predefined control policy [1, 2]. While traditional heuristic design methodologies were proven extremely useful in server configuration composed of a few, or a few of tens of servers, recent trends tend to create clouds composed of several millions of computing nodes. The complexity of such large-scale infrastructures prohibits the further use of traditional heuristic design methods, especially due to the extreme number of state variables, complex, stochastic, and non-linear interactions. In addition, the characteristics of typical infrastructures and applications are not explored yet deeply; accordingly, the collection of experimental data and checking the candidate control policies in an experimental environment is of an outmost importance.

My objective is the composition of a general purpose environment for data acquisition and experimental control policy development by creating a performance and availability control framework around Matlab, the leading-edge system identification and control implementation software. The measurement environment is based on standard system monitoring tools and a fault-injection engine.

## II. Infrastructure for monitoring and control

The *control framework* collects data from a set of *sensors* provided by platform- and application-specific measurement agents in system monitoring. This intelligent *decision-making* unit instructs system provisioning (as *actuator*) to meet or approximate optimization goals (set points). A pilot application infrastructure (Fig. 2) emulates a scaled-down datacenter, over which the framework measures and processes software and platform performance metrics in realistic scenarios. This pilot system adheres to widely-accepted standards and best practices, and is reconfigurable in runtime.

Nagios, a widely-used open source system monitor is used for application instrumentation, which collects measured data into a database of the central monitoring server. A Matlab program directly queries these data logs from the central database for post-processing, reaction planning, and execution

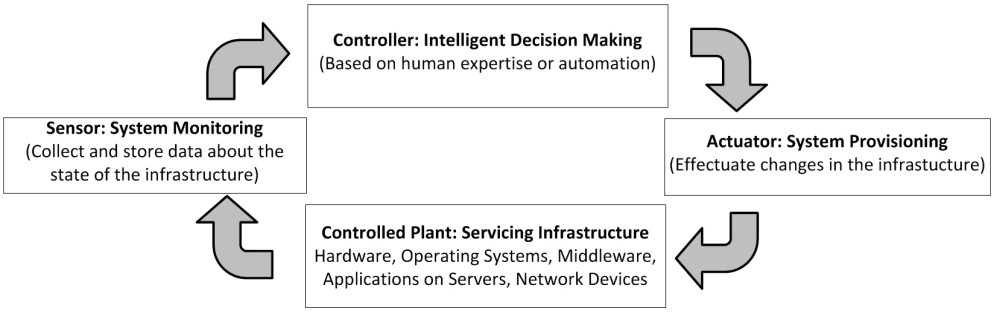



Figure 1: IT infrastructure management as a control loop

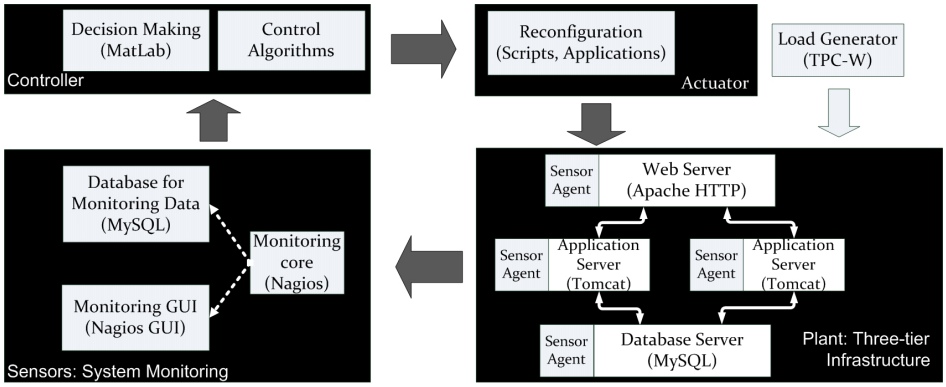



Figure 2: Three-tier infrastructure with monitoring and data processing

triggering. The three-tiered pilot TPC-W web benchmark infrastructure is composed of a web server, a runtime reconfigurable, load-balanced dual application server cluster, and a database server. A number of emulated browsers serve as workload generators according to the TPC-W specifications.

## III. Initial results on large-scale IT infrastructure control

Some core problems of large-scale IT infrastructure control were addressed at first: sensor selection for data processing [3], and large-scale monitoring- and experiment-based simulation [4], transaction tracking in datacenters for workload characterization [5], and applying control-schemes to lossless datacenter networks to achieve latency and robustness goals [6]. [3] compares different approaches to the problem of selecting a small subset, out of the huge set of sensors offered by monitoring tools, still faithfully characterizing the system for control purposes. We concluded that linear methods methods are more accurate in 'normal' states of operation, while entropy-based approximations yield better results in critical 'degrading' states (which is critical for predicting and possibly avoiding overload). [4] investigates the analysis of datacenter infrastructures by means of analytical and simulation methods, and we argue that rigorous datacenter design and control benefit from full-scale simulations already proven useful in high-performance computing [7], but require advanced monitoring and modelling.

## IV. Conclusions and future work

The presented experimental infrastructure for large-scale control development is functional. We will continue with detailed evaluation, system identification, and – on this basis – model-based control. Validation of the scalability of the empirical results gained in the framework to large-scale datacenters will be done by simulation-based model analysis [4, 7] parametrized by experimental data.

## References

[1] M. Parashar and S. Hariri, *Autonomic computing: concepts, infrastructure, and applications*, CRC Press, 2006.

[2] J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury, *Feedback control of computing systems*, IEEE, 2004.

[3] G. Paljak, I. Kocsis, Z. Egel, D. Toth, and A. Pataricza, "Sensor Selection for IT Infrastructure Monitoring," in *Third International ICST Conference on Autonomic Computing and Communication Systems*, 2009.

[4] M. Gusat, C. DeCusatis, C. Minkenberg, L. McKenna, K. Bhardwaj, G. Paljak, A. Pataricza, and I. Kocsis, "Benchmarking the Ethernet-Federated Datacenter," in *Data Center Converged, Virtual Ethernet Switching Workshop*, 2009.

[5] G. J. Paljak, "Transaction tracking in large scale datacenters," Tech. Rep. RZ3743, IBM Research Zurich, 2009.

[6] M. Gusat, C. Minkenberg, and G. Paljak, "Flow and congestion control for datacenter networks," Tech. Rep. RZ3742, IBM Research Zurich, 2009.

[7] C. Minkenberg and G. R. Herrera, "Trace-driven co-simulation of high-performance computing systems using omnet++," in *2nd International Workshop on OMNeT++ (hosted by SIMUTools 2009)*. ICST, 2009.

# MODELING OF HYBRID CONTROL SYSTEMS

**András VÖRÖS**
**Advisor: Tamás BARTHA**

## I. Introduction

Petri nets are widely used to model and analyze discrete event dynamic systems (DEDS). However, modern complex control systems consist of both continuous and discrete parts. Usually there is a discrete digital control component defining a mode of separation, and some kind of continuous controllers. These control systems are the so-called hybrid systems, describing the behavior of them we have to combine both continuous and discrete approaches. In this paper I introduce two Petri net based approaches for the modeling of hybrid control systems.

## II. Discrete and Continuous Petri Nets

Petri nets are graphical models for concurrent and asynchronous systems providing both structural and dynamical analysis possibilities like reachability and invariant behavior analysis.

A (marked) discrete Petri net is a 5-tuple $N = (P, T, w^-, w^+, M_0)$ represented graphically by a digraph, where $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places, $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions, $P \cap T = \emptyset$, $w^-$: $P \times T \rightarrow N$ is the input, $w^+$: $T \times P \rightarrow N$ is the output incidence function for each transition, represented by weighted arcs from places to transitions and from transitions to places, $w^*_{i,j}$ (or $w^*(p_i,t_j)$ ) is the signed weight of the arc between the $i$-th transition and the $j$-th place and $M_0 : P \rightarrow N$ is the initial marking, represented by tokens in the places. The $|P| \times |T|$ dimension matrix $W$ incidence matrix is built from the values of $w^*_{i,j}$ respectively (formally: $W = \| w^*_{i,j} \|$). With the help of this matrix we can write up the fundamental state equation: $M = M_0 + W \cdot s$, where $s$ is the firing vector corresponding to the transitions. Reachability: we call $M$ reachable from $M_0$ if exists a sequence $\sigma = \langle s_1, s_2 \ldots s_n \rangle$ of firings that $M_n = M_0 + W \cdot \sigma$ and $M \leq M_n$. The positive integer vector solution of this equation provides us a necessary condition for the reachability problem.

A timed Petri net has a timing function *time*: $T \rightarrow R^+$. This function associates a time duration to each transition, this duration is the time from the enabling of the transition to the firing.

Continuous Petri nets [1] can be gained by the *fluidification* of discrete nets. We say that a Petri net is continuous if the firings and the tokens in a place can not only be integer but real numbers. It is a relaxation of the discrete case, which can be gained by the split of each marking into $k$ tokens, where $k$ tends to infinity. In this case, the reachable markings become infinite.

A marked continuous Petri net is a 5-tuple $N = (P, T, w^-, w^+, M_0)$, P and T are similar to the discrete case, but $w^-$: $P \times T \rightarrow R^+$ is the input incidence function, $w^+$: $T \times P \rightarrow R^+$ is the output incidence function and $M_0$: $P \rightarrow R^+$ is the initial marking.

A transition $t$ is enabled at $M$ if for every $p_i$, where $w^-(p_i, t) > 0$, $M(p_i) > 0$.

The main advantage of the continuous relaxation of discrete nets is that it preserves the invariant behavior and *liveness* properties of the system in a significantly reduced state space.

Timing can be extended to the continuous case: we associate each transition a maximum firing speed, which is defined: if $d_i$ is the timing associated to $t_i$ in the discrete net, then $1/d_i$ will be the firing rate (flow through the transition) of the corresponding continuous net. This provides the modeling of processes, which are described by linear differential equations from any order.

Combining the approaches, we have to define the interface between the discrete and continuous nets. The discrete parts can affect the continuous parts through the continuous transitions as they can enable or disable them. If $n$ discrete tokens are needed to enable a continuous transition, after the firing of it we have to take $n$ token back to the discrete net. A continuous transition can't change the

number of discrete tokens, by definition [1]. The continuous part can also affect the behavior of the discrete part, but in this case it is not necessary, that the amount of token remains the same at the continuous net. This affect means an arc from a continuous place to a discrete transition with a given weight; this will always fire when it is enabled as the priority of discrete transitions is higher than continuous transitions [1]. This arc will indicate weather the continuous part reached a predefined state, which is important for the discrete controller parts and continuous transitions can also compute the output signal to the controlled system. With the help of this approach we can model the influence of the discrete controller to the continuous controller and to the controlled processes efficiently.

Reachability analysis in hybrid Petri nets differs from discrete ones, as we have to consider continuous trajectories too. In the case of autonomous hybrid Petri nets - where continuous parts are considered as a relaxation of discrete nets - the reachability space is *relaxed* as we take into account only the fact weather a continuous place is marked or not, without the exact marking.

In timed hybrid Petri nets we examine the invariant behavior (IB) states of the system, where IB-state is such that the marking of the discrete part and the instantaneous speed vector of the continuous part remain constant as long as the system is in the same IB-state [1] . Each IB-state corresponds to a state in the corresponding hybrid automata, so after the determination of the set of IB-states we are also able to construct hybrid automata from the Petri net.

## III. Object oriented approach

Hybrid models can be divided into smaller parts - objects – in order to make the analysis easier [2]. This approach defines a hierarchy in the system. As complex control systems consist of some smaller components, this decomposition method is a natural way to make the analysis easier. Each controller object is described by a Petri net, which uses continuous variables during the operation, and shares variables, places or transitions with the other objects. The creation of object oriented hybrid Petri net consists of a few steps: (1.) Definition of the control subsystems; (2.) Definition of the control Petri nets in each subsystem; (3.) Definition of the object variables; (4.) Definition of the interfaces between the objects, which contains shared variables and transitions; This few steps enable us to decompose the analysis into smaller steps and we can examine analysis questions locally. At first we examine the controller Petri nets of the object under examination with the help of linear logic. The smaller controller Petri nets are restricted to be safe. Linear logic let us make assumptions to the whole system from some components as linear logic makes the analysis process composable [2] .
The analysis contains the following steps: (1.) Formal specification of the property statement; (2.) Analysis of the first object: (2.1.) Analysis of the discrete dynamics with the help of linear logic; (2.2.) Analysis of the continuous dynamics for the necessary discrete states; (2.3.) Analysis of other object, which affects the analysis of the first object; The other objects affect only through the interface transitions and variables the behavior of the analyzed object; we have to examine other objects only when interface transition or variable is reached during the process. In each level of object hierarchy well defined interfaces help us to trace changes in the controller, and from the simple and small controller objects we are able to compose complex hybrid controller systems.

## IV. Conclusion, further work

Combining the above introduced methods may provide an efficient analysis method for hybrid systems. The next step will be implementing analysis algorithms and examine the adaptability of them to our industrial systems and problems.

## References

[1] René David, Hassane Alla: Discrete, Continuous and Hybrid Petri Nets, Springer 2005, ISBN: 3-540-22480-7

[2] E. Villani, P. E. Miyagi, R. Valette: Modelling and Analysis of Hybrid Supervisory Systems, Springer 2007