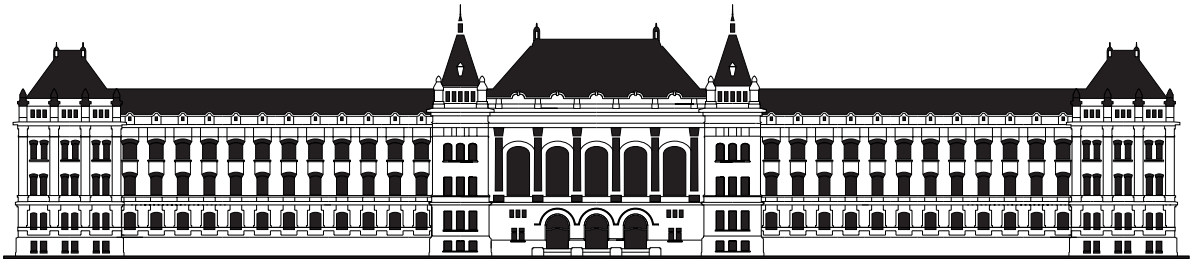
PROCEEDINGS
OF THE
15TH PHD MINI-SYMPOSIUM

FEBRUARY 4–5, 2008
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUILDING I



BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPARTMENT OF MEASUREMENT AND INFORMATION SYSTEMS

Head of the Department: Prof. Dr. Gábor PÉCELI

Conference Chairman:
dr. Béla PATAKI

Organizers:
Ákos HORVÁTH
Gergely KISS
Imre KOCSIS
Máté KOVÁCS
György OROSZ

Homepage of the Conference:
http://www.mit.bme.hu/events/minisy2008/

Sponsored by:
IEEE Hungary Section (technical sponsorship)
Schnell László Foundation
evosoft Hungary Kft.

ISBN 978-963-420-938-6

FOREWORD

This proceedings is a collection of the extended abstracts of the lectures of the 15th 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. 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 six main groups. These are Biomedical Measurement and Diagnostics; Embedded and Intelligent Systems; Fault-tolerant and Dependable 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 fifteen-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 Vrije Universiteit Brussel, Belgium; Aalborg University, Denmark; LAAS-CNRS, France; TU Darmstadt, Germany; University of Karlsruhe, Germany; University of Firenze, Italy; Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest; KFKI Research Institute for Particle and Nuclear Physics of the Hungarian Academy of Sciences, Budapest; Robert Bosch Kft., Hungary.

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 15, 2008.

Béla Pataki

Chairman of the PhD Mini-Symposium

LIST OF PARTICIPANTS

Participant	Advisor	Starting Year of PhD Course
ALTRICHTER, Márta	HORVÁTH, Gábor and ANDREOPOULOS, Bill	2005
BÓDIS-SZOMORÚ, András	DABÓCZI, Tamás and FAZEKAS, Zoltán	2005
BOKOR, Péter	PATARICZA, András	2005
COULIBALY, Sékou Tidiani	HORVÁTH, Gábor	2007
HAMAR, Gábor	HORVÁTH, Gábor; TARJÁN, Zsuzsanna and VIRÁG, Tibor	2005
HORVÁTH, Ákos	VARRÓ, Dániel	2006
HULLÁM, Gábor	ANTAL, Péter and STRAUSZ, György	2005
JUHÁSZ, Sándor	HORVÁTH, Gábor	2007
KISS, Gergely	FEHÉR, Béla	2006
KOCSIS, Imre	PATARICZA, András	2006
KOVÁCS, Máté	MAJZIK, István	2006
KRÉBESZ, Tamás István	KOLUMBÁN, Géza	2007
LASZTOVICZA, László	PATAKI, Béla	2003
MICSKEI, Zoltán	MAJZIK, István and WAESELYNCK, Hélène	2005
NEPUSZ, Tamás	STRAUSZ, György and BAZSÓ, Fülöp	2005
OROSZ, György	SUJBERT, László and PÉCELI, Gábor	2006
PÁSZTOR, Péter	PATARICZA, András	2006
RAIKOVICH, Tamás	FEHÉR, Béla	2007
RÁTH, István	VARRÓ, Dániel	2006
SCHERER, Balázs	HORVÁTH, Gábor	2007
SISAK, Áron	MAJZIK, István	2007
TÓTH, Dániel	PATARICZA, András	2006

Program of The MINI-SYMPOSIUM

Information Mining and Knowledge Representation

Conference Schedule

Time	February 4, 2008	Time	February 5, 2008
8:30	Conference Opening Opening Speech: Gábor Péceli Evosoft presentation	8:30	Embedded and Intelligent Systems
8:40	Measurement and Signal Processing		
10:00	Fault-tolerant and Dependable Systems	10:00	Information Mining and Knowledge Representation
Launch break			
13:30	Biomedical Measurement and Diagnostics		
14:45	Model-based Software Engineering		

CAMERA CALIBRATION AND POSE ESTIMATION USING PLANAR FEATURES AND SENSITIVITY ANALYSIS

András BÓDIS-SZOMORÚ
Advisors: Tamás DABÓCZI (BUTE-MIT), Zoltán FAZEKAS (MTA SZTAKI)

I. Introduction

In 3D computer vision, a camera model is a geometrical model that maps points of a 3D scene to corresponding image points; the latter being given in pixel coordinates. Many machine vision applications such as 3D object reconstruction, augmented reality (AR), visual servoing of robots or man-machine interfaces rely on the knowledge of this mapping [1].

In order to make use of a camera model, all or at least some of its parameters are usually calibrated off-line as a preliminary step using known world-to-image correspondences of some specific detectable features, typically corners, circular patches or linear segments.

Many object and camera tracking applications make use of the a priori knowledge of the intrinsic camera parameters and continously re-compute the extrinsic parameters over time [1]. In this case, referred as pose estimation, only the six degrees-of-freedom (DoF) of the Euclidean transformation between camera and world needs to be recovered. The precision of pose estimation in object tracking or the accuracy of a reconstructed object geometry in 3D reconstruction always coheres intensely with the accuracy of the preliminary calibration, thus highlighting the importance of an accurate calibration and a comprehensive sensitivity analysis.

This paper addresses preliminary camera calibration from static and planar features including a sensitivity analysis. Significant modifications are proposed herein to some existing methods. Two very practical applications of the proposed methods are outlined in brief. The aforementioned modifications and methods presented here are embodied in two recently developed Matlab GUI Toolboxes that are going to be freely available.

II. Calibration using planar features

A. Camera models and parameters

A projective (pinhole) camera can be represented as a linear mapping $\mathcal{P}$ from the projective space $\mathbb{P}^3$ to $\mathbb{P}^2$. It maps a 3D world point $\tilde{W} \in \mathbb{P}^3$ to its 2D image $\tilde{I} \in \mathbb{P}^2$: $\lambda\tilde{I} = P\tilde{W}$ where P is the 3×4 homogeneous camera matrix representation of $\mathcal{P}$ and λ is an arbitrary scale factor. In the inhomogeneous representation $W \in \mathbb{R}^3$ and $I \in \mathbb{R}^2$, the projective camera model becomes non-linear. It should be noted that lens distortions affect the inhomogeneous coordinates of the projected points. Therefore, when such distortions are considered, it is necessary to use the inhomogeneous representation. The mapping $\varphi\colon W \mapsto I$ in its general form can be written as

$$I = \varphi(\mathbf{p_{int}}, \mathbf{p_{ext}}, W), \tag{1}$$

where $\mathbf{p_{int}}$ is the vector of intrinsic parameters and $\mathbf{p_{ext}}$ is the 6-vector of extrinsic parameters representing a 6-DoF Euclidean transformation of the points from the world to the camera reference frame. This transformation is usually represented by a 3×3 rotation matrix $R = \begin{bmatrix}\mathbf{r_1} & \mathbf{r_2} & \mathbf{r_3}\end{bmatrix}$ and a translation vector $\mathbf{t}$. Since R has 9 elements but only 3-DoF, it is an over-parametrization of the model that involves 6 additional non-linear constraints induced by $R^TR = E$, where E represents the unit matrix. This is the reason why the Rodrigues-vector, quaternions or Euler-angles are commonly used to represent rotation [1]. Using the Rodrigues-vector $\mathbf{r} \in \mathbb{R}^3$, $\mathbf{p_{ext}} = (\mathbf{r}^T, \mathbf{t}^T)^T$.

B. *Camera parameters from planar patterns*

In case of camera calibration I_i and W_i $(i = 1, 2, \ldots, n)$ correspondences are known and the parameter vector $\mathbf{p} = (\mathbf{p_{int}}^T, \mathbf{p_{ext}}^T)^T$ is to be determined. Since planar fidutials are easier to create than 3D ones, recent methods prefer using planar patterns. In the planar case, without loss of generality $W_i = (X_i, Y_i, 0)^T$ and the homogeneous mapping can be re-written as $\lambda \tilde{I}_i = \lambda(u_i, v_i, 1)^T = H(X_i, Y_i, 1)^T$; where $H = \begin{bmatrix}\mathbf{h_1} & \mathbf{h_2} & \mathbf{h_3}\end{bmatrix} = K \begin{bmatrix}\mathbf{r_1} & \mathbf{r_2} & \mathbf{t}\end{bmatrix}$. Here H is a 3×3 homography matrix mapping all W_i to I_i, and K is the 3×3 camera calibration matrix depending solely and linearly on the intrinsic parameter vector $\mathbf{p_{int}}$. Unfortunately, from a single planar arrangement it is impossible to recover all the 11 camera parameters, (i.e. the 6 extrinsic and 5 intrinsic parameters in case of no distortions). In [2], Z. Zhang presents an excellent solution to this problem by showing the same *known pattern* in m different *unknown orientations* to the camera. (Figure 1).

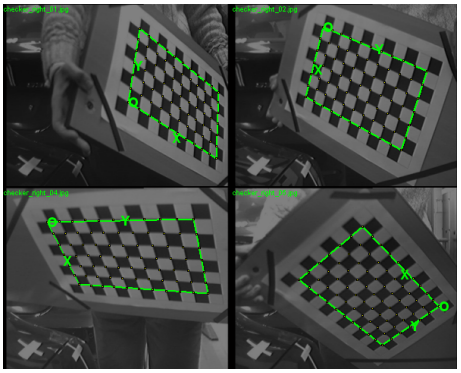


Figure 1: Calibration from planar patterns with our Matlab Toolbox

For each view, there is an individual $\mathbf{p_{ext,j}}$ $(j = 1, 2, \ldots, m)$ vector of extrinsic parameters while the intrinsic parameters are common for all views. After computing the homographies H_j from the point-correspondences, based on the orthonormality of $\mathbf{r_{1,j}}$ and $\mathbf{r_{2,j}}$ (see the constraints on H above), Zhang managed to build up a set of equations involving only the intrinsic parameters. These equations are linear in the elements of $B = K^{-T}K^{-1}$ and consequently non-linear in those of $\mathbf{p_{int}}$. First, B is computed by an LS method, and then K and $\mathbf{p_{int}}$ are recovered using non-linear formulas. From the individual H_j computed beforehand, all $\mathbf{p_{ext,j}}$ can be recovered. In other words, the pose estimation problem is solved for each of the m planes. The resulting camera parameters packed in $\bar{\mathbf{p}} := (\mathbf{p_{int}}^T, \mathbf{p_{ext,1}}^T, \ldots, \mathbf{p_{ext,m}}^T)^T$ are then refined by minimizing the reprojection error $C(\bar{\mathbf{p}}) = \sum_{i,j} d^2(I_{ij}, \hat{I}_{ij}) = \sum_{i,j} \|I_{ij} - \varphi(\mathbf{p_{int}}, \mathbf{p_{ext,j}}, W_{ij})\|^2$, where d denotes Euclidean distance, $\|.\|$ denotes vector 2-norm, I_{ij} is the i-th image point detected by a feature detector on the j-th image, $\hat{I}_{ij}$ is the reprojection of the corresponding 3D point W_{ij} onto the image plane using the camera model (1).

C. *Sensitivity analysis*

If the distribution of feature localization is a (2D) isotropic Gaussian with zero mean for all the images, and the world points W_i are precisely known, then the cost function minimized is a maximum likelihood (ML) cost function. Since the residual errors are realizations of the above-mentioned Gaussian noise, the standard deviation $\hat{\sigma}$ of all the residual coordinates is a good measure of the uncertainity in the image. When searching for the uncertainty of the camera parameters, one can back-propagate $\hat{\sigma}$ to the camera parameters through the camera model linearized at each reprojected point: $C_p = (J^T J)^{-1}\hat{\sigma}^2$, where C_p is the estimate of the $(9 + 6m) \times (9 + 6m)$ covariance matrix of all the parameters, $J = \frac{\mathrm{d}\hat{I}}{\mathrm{d}\bar{p}}$ is the stacked $2n \times (9 + 6m)$ Jacobian of the mapping evaluated at the optimal parameters and $\hat{I}$ is the vector containing all the reprojected image points $\hat{I}_{ij}$, thus having $2mn$ elements. Note that the hyper-matrix J is a sparse matrix. This is due to the fact that small perturbations in the orientation of the i-th plane affect only the reprojected points in the i-th image, though perturbations in the intrinsic parameters affect all. Finally, the deviation of the k-th camera parameter $k = 1, 2, ..., (9 + 6m)$ is estimated as $\hat{\sigma}_{p,k} = \sqrt{C_{p,kk}}$. It implies that uncertainty of the intrinsic parameters is considered independent of that of the extrinsic parameters.

A Toolbox for Matlab implementing a calibration similar to Zhang's method already exists at Caltech [3]. It is open-source and performs sensitivity analysis, as well. They back-propagate the measurement noise to all the parameters as discussed above and associate $\pm 3\hat{\sigma}_{p,k}$ uncertainty to the parameters corresponding to the 99.7% confidence level.

III. Problems and solutions

A. *Enforcing values for a subset of parameters*

In situations when exact a priori information about the mapping is available (e.g. the pixel aspect ratio is exactly known from the datasheet of the camera), it is desirable to enforce some parameters manually. Therefore, we extended the method of Zhang to handle cases when a subset of intrinsic parameters is fixed. It is important to note that the matrix B estimated with Zhang's linear method has 9 elements but only 6 or less DoF. This means that measurement noise may jeopardize the estimation in some cases. This threat increases when certain parameters are enforced. A subset of intrinsic parameters can be easily fixed using our Toolbox.

B. *Perspective-n-point problem with error in the world points*

The cost function used earlier expresses errors only in the image. It is not ML any more when significant errors are present in W_i. A practical example is when one may use Zhang's method for intrinsic calibration and an additional plane with noisy world points for a final pose estimation. Therefore, we derived the ML cost function for this case. Here, we supposed that measurement uncertainities of the 3D coordinates W_i are Gaussian and are independent of the errors in feature localization in the image. This condition is generally met in practice. The resulting cost function is:

$$C(\mathbf{p_{int}}, \mathbf{p_{ext}}) = \sum_i \|I_i - \hat{I}_i\|^2_{C_{Ii}} + \sum_i \|W_i - \hat{W}_i\|^2_{C_{Wi}}, \tag{2}$$

where $\|x\|^2_C$ denotes the Mahalanobis distance $(x^T C^{-1} x)$. Equation (2) introduces the 3D measurement errors $(W_i - \hat{W}_i)$ where $\hat{W}_i$ is the expected 3D location of W_i, while W_i is the measured one. The introduction of $\hat{W}_i$ means that the 3D points are considered as parameters and thus our parameter space is of dimension $(9 + 6m + 3n)$. Just for comparison, we also used a convenient approximation for the cost function (2) in which each $\hat{W}_i$ is estimated as the orthogonal projection of W_i to the ray through the image point I_i. This reduces the dimension of the parameter space back to $(9 + 6m)$ and gives an estimate very close to the ML one.

Also, the covariance matrices C_{Ii} and C_{Wi} appeared in (2). The first caracterizes the error of the feature detection in the image, the second, the localization errors in 3D. Therefore, this solution is only practical when C_{Ii} and C_{Wi} can be estimated. $C_{Ii} = diag(\sigma^2, \sigma^2), i = 1, 2, \ldots, n$ used to be a practical simplification. When $\mathbf{p_{int}}$ is preliminarily estimated e.g. by Zhang's method, the problem reduces to pose estimation, that is, estimating only $\mathbf{p_{ext}}$, that is, only 6 DoF needs to be recovered. A single planar arrangement with 4 points or more in general planar configuration is enough to compute an initial guess for $\mathbf{p_{ext}}$. This is called Prespective n-Point problem (PnP) for $n > 3$ [1]. The method is also extended to the multi-camera case where the 3D structure $(\hat{W}_i)$ is common for all the views.

C. *Modifications to the sensitivity analysis*

In the discussed sensitivity analysis, the covariance of the errors in the image is back-propagated to all the parameters. However, when *using* the results of the calibration, generally only one reference view is important and only the corresponding sub-matrix of C_p needs to be considered.

The situation gets more complicated when errors are present in the world points of one of the calibration planes. In our Toolboxes, we use checkerboard planes with world points considered exact and an additional reference plane with errors in the world points for pose estimation. The checkerboard-based calibration and the pose estimation to the reference plane are performed in two independent steps with independent measurements. Assuming Gaussian distribution for all the measurements, we extended the sensitivity analysis for this case. Since the ML cost function introduces the estimate of the expected location of the world points as well, our overall sensitivity analysis gives the covariance of the intrinsic parameters, of all the $6m$ extrinsic parameters with respect to m checkerboards plus 6 with respect to the reference plane and of the 3D location estimates of the world points in the reference plane.

IV. Applications and results

A. Calibration of a multi-camera corneal topographer

We used our Toolbox implementing the proposed methods to calibrate the multi-camera corneal topographer described in [4]. Calibration is done by rotating a planar pattern of size 1.2×1.2 cm at the expected location of the patient's cornea. The key problem experienced during the calibration was that due to the high focal length, the perspective effect in the images taken was neglectable and therefore a high uncertainity was present in the principal point estimation. This was solved by fixing the principal point in the image center. An alternative solution is to use an affin camera model. For this purpose, the specular surface reconstruction algorithm has been adapted for an affine model, as well.

B. Calibration of a stereo lane detection system

Another interesting application of our methods is the calibration of the stereo lane detection system described in [5]. Intrinsic calibration of the cameras fixed to the vehicle's chassis is performed with a hand-held and rotated pattern (see Figure 1). An additional pose estimation is also required relative to the road. For this purpose, markers were placed in front of the vehicle and their 3D positions were measured. The measurement covariances C_{Ii} and C_{Wi} were estimated, as well. Because the 3D points W_i were measured with high uncertainty and localization errors in the image were relatively small, the proposed cost function gave significantly better results at reprojection when refining the 3D points $\hat{W}_i$ as proposed (Figure 2). Both the simplified mono version with reduced parameter space and the stereo version of the algorithm gave qualitatively the same result as shown on the right side of Figure 2. This proves the effectiveness of the approach. The estimation algorithms for an additional plane (the road's plane) are implemented in a second Toolbox supporting multiple cameras and based on the first one.

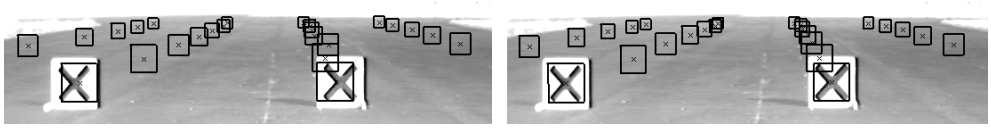


Figure 2: Pose estimation: markers reprojected using the optimal parameters when minimizing errors in the image only (left) and when using the proposed co-minimization based on (2) (right).

V. Conlclusions and future work

We proposed some important modifications to existig methods for calibrating a single or multiple perspective cameras to planes, including sensitivity analysis, parameter fixation, and errors present in the world point locations. The proposed methods are implemented in two Matlab GUI Toolboxes and have been used in two completely different multi-camera applications with success. Further work consists of a specific calibration and reconstruction method for the corneal topographer and on-line pose estimation for the lane detection system under development.

References

[1] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[2] Z. Zhang, "A flexible new technique for camera calibration," Tech. Rep., Microsoft Research, Microsoft Corporation, Aug. 2002, URL: http://research.microsoft.com/∼zhang.

[3] J.-Y. Bouguet, *Camera Calibration Toolbox for MATLAB*, California Institute of Technology, Apr. 2007.

[4] A. Soumelidis, Z. Fazekas, Z. Lichtenberger, and A. Bódis-Szomorú, "Embedded computing solutions used in a multi-camera corneal topographer," in *Regional Conference on Embedded and Ambient Systems (RCEAS)*, 2007, to be published in feb. 2008.

[5] A. Bódis-Szomorú, Z. Fazekas, and T. Dabóczi, "High-accuracy calibration of lane detection systems using stereovision," in *Instrumentation and Measurement Technology Conference (IMTC)*, 2006, ISBN: 1-4244-1080-0.

CONSIDERATIONS ON ACOUSTIC LOCALIZATION WITH DISTRIBUTED WIRELESS SENSOR NETWORKS

Gergely KISS
Advisor: Béla FEHÉR

I. Introduction

Abstract – Finding the spatial location of an acoustic source is a challenging area of digital signal processing. Localization of sources has been applied to different fields including ecological (animal tracking), civil (traffic monitoring), medical (ultrasound) and military applications (radar, sonar, vehicle tracking, and sniper localization). The advancements in hardware design and technologies allow for sensor-level implementation of more sophisticated algorithms. Specifically, wireless sensor nodes can benefit from these miniaturized but highly integrated systems as they can significantly enhance the computational capabilities of each node. In this paper an outlook on the available systems for and the most common issues with acoustic localization is provided.

II. The common approach to localization

The frequency spectra of acoustic sources typically lie within the 20 Hz – 20 kHz range. When localizing, knowing the signal frequency is important since it has to be inferred in geometry setup and algorithm design. Assuming the sound propagates at constant speed from the source concentrically, the wavefront reaches the sensing elements (microphones) of other nodes at different times [1]. It is possible to locate an acoustic source based on these delay values. However, there are several effects which cause difficulties in accurate localization due to physical backgrounds of sound propagation, including echoes, reverbs, ground reflection, etc.

III. Existing hardware platforms

Numerous attempts have been made to design and develop wireless sensor network systems highly suitable to accomplish acoustic source localization [2][3]. These systems use FPGAs as their primary processing unit supported by other components such as multiple analog input channels, memories, communication interfaces, and on-board radio. They usually rely on and expand the capabilities of existing wireless sensor nodes, such as Crossbow MicaZ or Telos-B [5]. The idea behind using high performance nodes is to reduce the amount of data to be transmitted among nodes or to the base station to optimally utilize the bandwidth. These systems are typically used to perform acoustic transient localization and/or vehicle tracking outdoors [2][3][4][7].

IV. Source and node localization

In order to perform acoustic localization with distributed wireless sensor networks, the location of each node should be known.

Self localization – or node localization – can be performed in different ways. The most common methods use RF as a support for acoustic localization since delays are not introduced in the range of acoustic domains. The sensor nodes are capable of determining their relative locations [8]. In some cases, for example when localizing indoors this is sufficient. However, when absolute position is a requirement the use of anchor nodes in inevitable and makes it a more difficult scenario. Deployment cannot be carried out random as at least some of the nodes should be equipped with GPS units or should be placed at known positions and within range at the same time.

Source localization can either be active or passive depending on whether the object of interest emits sound itself or has to be triggered in some way. We can differentiate between transient and continuous localization – tracking – tasks. In the first case the localization can be based on the onset and end detection. This is not true for the second case since no onset and end of signal are available, extraction of features or usage of beamforming-like algorithm is required.

V. Difficulties with cross-correlation

Assume the case when all the nodes are already deployed, the positions are known and passive transient localization is to be performed. Measurements show that in some cases cross-correlation based TDOA estimates are not accurate due to physical principles of wave propagation. The ground reflections and lack of coherency distorts the measurements. Application of matched filtering would have the same problem. Applying the cross-correlation on the resampled version of time series shows the peak correlation coefficient are at resample ratios of a few percents. This phenomenon could also result from nonlinear behavior of microphone in case of very high amplitude excitation.

VI. Node collaboration

Localization needs the wireless sensor network nodes to be synchronized. A beacon based global timestamping which is simple and easy to implement yet might be adequate for the purpose. Other methods like precision time synchronization protocol (PTP, IEEE1588) can be used. PTP takes into account compensates for the transfer delays.

A crucial element of each sensor network is communication among nodes. The network bandwidth provided by the almost industry-standard Zigbee (IEEE 802.15.4) wireless personal area network can be insufficient and not intended to be used for high volume time series transfers. One option is to use sensor-level time series compression which can be fine-tuned being signal characteristics know, or sensor-level feature extraction. Both of these requires the nodes to be capable of fast computation since huge amount of time required for preprocessing is disadvantageous for real-time applications.

Wireless sensor networks always face the problem of low power versus high computational capability. An optimum solution can be found using the on-board power management features of the board.

VII. Indoor challenges

Indoor localization faces much more issues than outdoor localization because of the following phenomena [9].

Reverberation is unavoidable at indoor localization [6]. The source signal is reflected from the surface of different objects which results in a complex signal containing the direct path and its delayed and attenuated copies, it can be written as:

$$x(t)=\sum_{k=0}^{L-1} a_k s(t-t_k)+n(t)\,. \qquad (1)$$

where $s(t)$ is the source signal, a_k is an attenuation coefficient, $n(t)$ is additive Gaussian white noise and L is the number of multipath components. It can be seen, that apart from the direct path delay, several other delays appear which leads to the necessity of sophisticated algorithms capable of separating direct path from reverberations. There is also a chance that an obstacle prevents direct wave propagation from source to the sensor which results in non-line of sight problem and makes it difficult to create a measure for reliability.

Most localization methods are TDE (Time Delay Estimation) algorithms, based on TOAs (Time of Arrivals) and TDOAs (Time Difference of Arrivals). The first refers to the time at which a defined event happens at the sensor side with global clock reference, the second is the difference among the first ones. The propagation speed of sound is inherent when working with TOAs and TDOAs.

Temperature changes affect localization accuracy. Temperature change can be a disadvantageous effect at indoor setups as well. The temperature dependency can be written as [9]:

$$c(\theta) = 331.4\sqrt{1+3.66\cdot 10^{-3}\cdot\theta} \approx 331.4 + 0.6\cdot\theta\,. \qquad (2)$$

where c is measured in $\frac{m}{s}$, θ is in °C. For example assuming 10 meters sensor separation 30 degrees difference of temperature leads to 60 cm ambiguity on a single TOA or TDOA, which error accumulates at localization resulting in remarkable error. The microphone directivity also becomes an issue at higher frequencies [9].

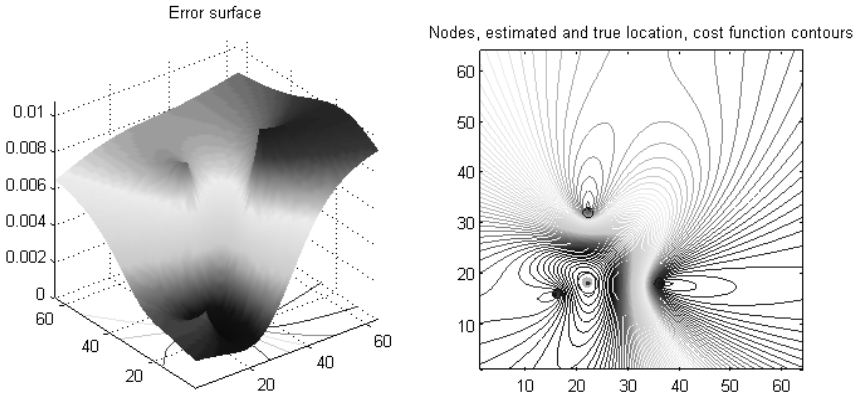


Figure I: TDOA error surface, sensor network of three nodes (black dots), right: contours of the surface projected on the *x*-*y* axis (distance)

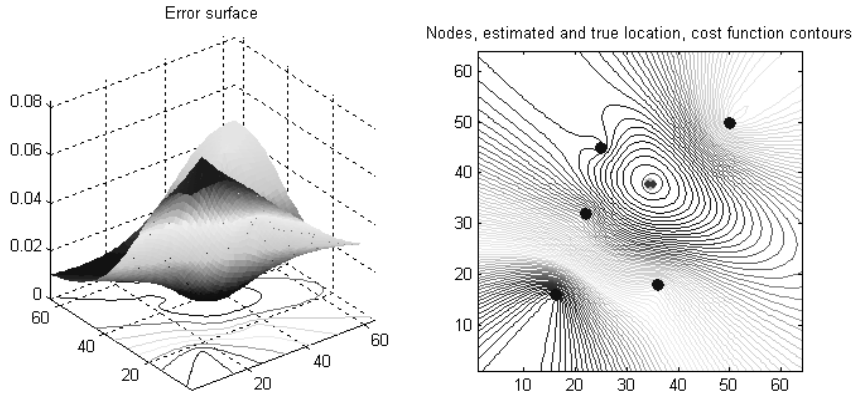


Figure II: TDOA error surface, sensor network of five nodes, contours are concentric circles around localization source (minimum of the surface)

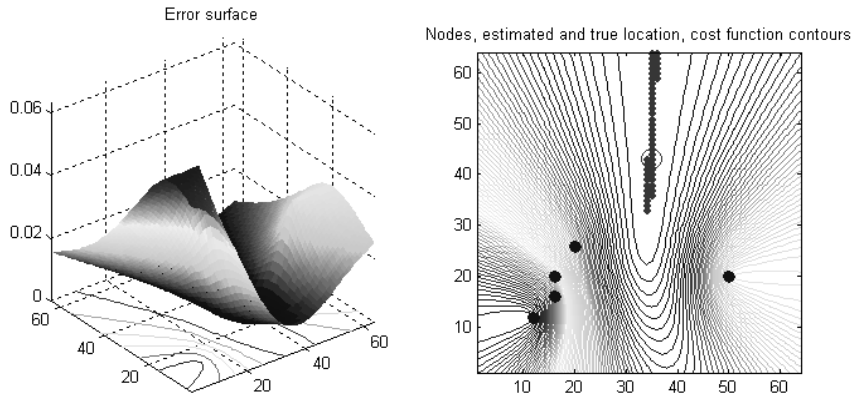


Figure III: Disadvantageous arrangement of nodes (big black dots), the minimum is not a distinct point but spans over a huge region (small black dots forming a vertical line in the middle)

VIII. Localization with accurate TDOA estimates

Consider a distributed wireless sensor network consisting of several nodes deployed at known positions each of them equipped with microphones. Simulations show despite having fairly accurate TDOA estimates localization performance can be poor.

The true TDOAs are calculated for an object moving on a specified trajectory. Then, with the knowledge of these a surface (S) is created on a predefined grid in the proximity of the sensor network. Point $P(x, y) \in S$ shows what would be the error of TDOA, if *(x,y)* were chosen as the coordinates of the localizable source. The shape of S visualizes the achievable localization performance. As the object keeps on moving S represents how well the location can be determined based on the true TDOAs. The simulation (Fig I.) shows angular accuracy (direction of arrival estimation) to appear much better than radial (ranging). Usage of more sensor network elements can overcome the issue, but far from the network the effect is inevitable.

The sensor network on both Fig. II. and Fig. III. consists of five nodes although in different geometry. The first one provides a good minimum at the true location resulting in accurate localization. The second can only determine the direction, the range information is ambiguous. This visualization technique helps finding the weak points of different arrangements.

IX. Conclusion

Crucial issues on acoustic localization with distributed wireless sensor network were provided along with available systems and error analysis of TDOA based localization. Further studies include a detailed description of the developed system along with the error analysis in several distinct measurement environment conditions. Ideas on how to optimize algorithms for robustness against reverberation and echoes will be provided.

References

[1] Gergely Kiss, “Acoustic Design of FPGA-based system supporting acoustic source localization”, Students’ Scientific Conference, BUTE 2005.

[2] Á. Lédeczi, G. Kiss, B. Fehér, P. Völgyesi, Gy. Balogh, “Acoustic Source Localization Fusing Sparse Direction of Arrival Estimates”, Proc. of Fourth Workshop on Intelligent Solutions in Embedded Systems (WISES’06), Vienna, Austria, June 30th 2006

[3] M. R. Azimi-Sadjadi, G. Kiss, B. Fehér, S. Srinivasan, Á. Lédeczi: Acoustic Source Localization with High Performance Sensor Nodes, Proc. of SPIE’07 Defense & Security Symposium, 8-13th April 2007, Orlando, FL, USA, Vol. 6562. pp. 65620Y.

[4] M. R. Azimi-Sadjadi, Y. Jiang, G. Wichern, Properties of Randomly Distributed Sparse Acoustic Sensors for Ground Vehicle Tracking and Localization, Proc. of the SPIE'06 Defense and Security Symposium, Vol. 6201, Orlando, FL, April 2006.

[5] Crossbow Technology Inc. http://www.xbow.com/Products/productsdetails.aspx?sid=62

[6] T. Gustafsson, B. D. Rao, M. Trivedi, Source Localization in Reverberant Environments: Modeling and Statistical Analysis, IEEE Transactions on Speech and Audio Processing, Vol. 11, No. 6, November 2003

[7] Lédeczi Á., Völgyesi P., Maróti M., Simon Gy., Balogh Gy., Nádas A., B. Kusy, Sebestyén Dóra, Pap G.: Multiple Simultaneous Acoustic Source Localization in Urban Terrain Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, 37235, USA

[8] M. Maroti, B. Kusy, G. Balogh, P. Volgyesi, K. Molnar, A. Nadas, S. Dora, and A. Ledeczi, “Radio interferometric positioning,” Proc. of ACM Third International Conference on Embedded Networked Sensor Systems (SENSYS’05), November 2005

[9] H. Farrokhi, R. J. Palmer, “The Designing of an indoor acoustic ranging system using the audible spread spectrum LFM (chirp) signal” Canadian Conference on Electrical and Computer Engineering, May 2005

INTRODUCTION TO SIGN ERROR SPECTRAL OBSERVER

György OROSZ
Advisors: László SUJBERT, Gábor PÉCELI

I. Introduction

The sign error observer algorithm introduced in the paper is based on the relationship between the least mean square (LMS) [1] and resonator based observer algorithms [2][3], but utilizes the sign error LMS algorithm [4] for estimating the state variables of the observed system. Since the algorithm uses the signum of the error of the estimation, significant reduction in the amount of data required for the algorithm and in the computational demand can be achieved. Hence the utilization of this algorithm reduces design restrictions in systems with limited resources (e.g. bandwidth of communication channels).

Possible utilizations of the observer are the Fourier decomposition of signals and adaptive control. An application example is an active noise control (ANC) system [8] that uses wireless sensor network (WSN) for noise sensing [6]. This is a straightforward field for the deployment of this algorithm, since ANC systems require lots of sensors and relatively high sampling frequency taking into account the typical bandwidth of the WSN's radio standards (e.g. ZigBee) and the real time data transmission, so data reduction plays important role.

II. Review of the traditional resonator based observer structure

The resonator based observer was designed to follow the state variables of the so-called conceptual signal model [2]. The signal model is described as follows:

$$\mathbf{x}_{n+1} = \mathbf{x}_n\ ; \qquad \mathbf{x}_n = [x_{i,n}]^{\mathrm{T}} \tag{1}$$

$$y_n = \mathbf{c}_n \cdot \mathbf{x}_n = \sum_{i=-L}^{L} c_{i,n} \cdot x_{i,n} \tag{2}$$

$$\mathbf{c}_n = [c_{i,n}];\ c_{i,n} = e^{j \cdot \omega_i \cdot n} = e^{j \cdot i\omega_1 \cdot n},\ i = -L \ldots L\ , \tag{3}$$

where $\mathbf{x}_n$ is the state vector of the signal model at time step n, y_n is its output (the input of the observer), $\mathbf{c}_n$ represents the basis functions. To generate a real signal $\omega_{-i} = -\omega_i$ shall be satisfied. Obviously, in these cases the corresponding state variables shall form complex conjugate pairs. The conceptual signal model can be considered as a summed output of resonators which can generate any multisine with components up to the half of the sampling frequency. The corresponding observer is (see Fig. 1):

$$\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n + \mathbf{g}_n\,(y_n - \mathbf{c}_n \cdot \hat{\mathbf{x}}_n) = \hat{\mathbf{x}}_n + \mathbf{g}_n\,(y_n - y'_n) = \hat{\mathbf{x}}_n + \mathbf{g}_n\, e_{\mathrm{n}}; \qquad \mathbf{g}_n = [g_{i,n}]^{\mathrm{T}} = [r_i\, c^*_{i,n}]^{\mathrm{T}}, \tag{4}$$

where $\{\hat{\mathbf{x}}_n = [\hat{x}_{i,n}]^{\mathrm{T}};\ i = 1 \ldots N;\ N = 2L+1\}$ is the estimated state vector, $\{r_k;\ k = 1 \ldots N\}$ are free parameters to set the poles of the system, and * denotes the complex conjugate. N is the number of harmonic components. Due to the complex exponentials, the channels of the observer can be considered as time-invariant systems with a pole on the unit circle. This is why they are called resonators. If the resonator poles are arranged uniformly on the unit circle, and $\{r_k = {}^1/_{\mathrm{N}};\ k = 1 \ldots N\} \rightarrow \mathbf{g}_n = {}^1/_N\ \mathbf{c}_n^{\mathrm{H}}$ ($^{\mathrm{H}}$ denotes the conjugate transpose), the observer has finite impulse response, and the observer corresponds to the recursive discrete Fourier transform (RDFT) [2]. If the alignment of the resonators is not uniform, the settling is no longer deadbeat, but the system is still stable.

Since (4) corresponds to the formula of LMS, (4) can be interpreted as the state variables were updated by the complex LMS algorithm, where the reference signal is $\mathbf{c}_n$. Using this relationship

between the observer and LMS [3] in the proposed new observer structure the sign error LMS (SE-LMS) algorithm is used for updating the state variable $\hat{\mathbf{x}}_n$.

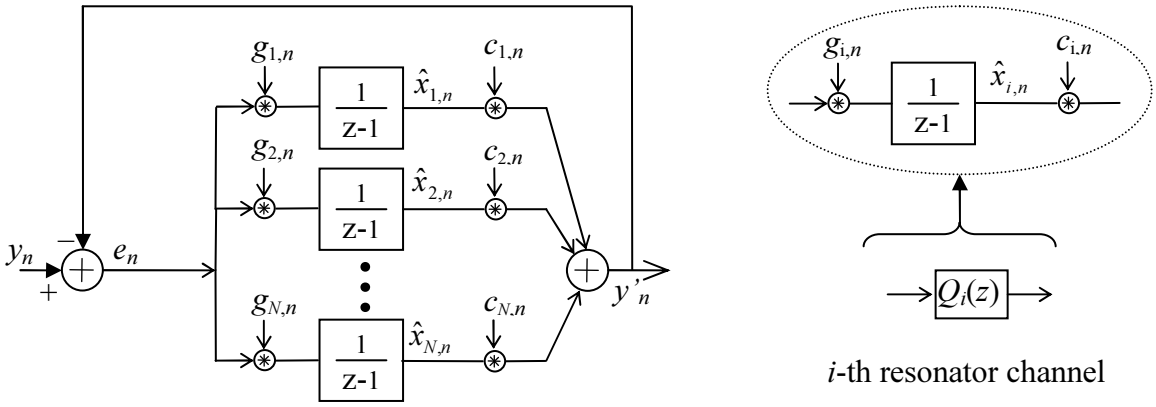


Figure 1: Basic configuration of the resonator based observer

III. The Sign Error Observer Structure

The proposed sign error structure can be seen in Fig. 2. The update procedure is the following:

$$\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n + \mathbf{g}_n \operatorname{sgn}(e_n); \qquad \mathbf{g}_n = [g_{k,n}]^T = [\alpha c^*_{k,n}]^T = \alpha \mathbf{c}_n^H, \tag{5}$$

where $e_n = (y_n - y'_n)$ is the error of the estimation. $\operatorname{sgn}(x)=|x|/x$, i.e. $\operatorname{sgn}(x)=+1$ if $x>0$, -1 if $x<0$ and $\operatorname{sgn}(x)=0$ if $x=0$. It means that $v=1$ in Fig. 2 in the case of this simple sign error observer. This updating requires only the knowledge of the sign of the error, so it needs less computation, than the original algorithm—see (4)—, and the amount of data required for the operation is reduced. This is advantageous if it is implemented in systems with constrained resources. α is used for setting the transient and steady state behavior of the observer.

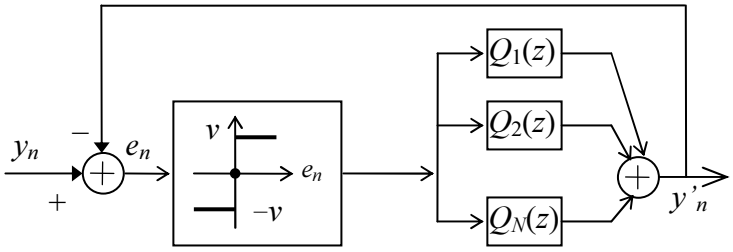


Figure 2: Basic configuration of the resonator based sign error observer

The steady state error of the observer can be determined by adapting the results in [4] for this structure:

$$E_a(n) = \frac{1}{n}\sum_{k=0}^{n-1}|e_k| \;\le\; \frac{\|\mathbf{x}\|^2}{2\alpha n} + \frac{N}{2}\alpha, \tag{6}$$

where E_a is the absolute mean error. (6) implies that if $n\to\infty$ (system is in steady state), the average absolute error is bounded by $N\alpha/2$ that is proportional to the convergence parameter α.
The settling time M of the observer can be estimated by the recursive expansion of (5):

$$\hat{\mathbf{x}}_M = \sum_{j=0}^{M-1} \alpha \cdot \mathbf{c}_j^H \cdot \operatorname{sgn}(e_j) + \hat{\mathbf{x}}_0 \,. \tag{7}$$

Taking the absolute value, and assuming that the initial state $\hat{\mathbf{x}}_0 = 0$ we get:

$$\left\|\hat{\mathbf{x}}_M\right\| = \left\|\sum_{j=0}^{M-1} \alpha \cdot \mathbf{c}_n{}^{\mathrm{H}} \cdot \mathrm{sgn}(e_j)\right\| \le \sum_{j=0}^{M-1} \left\|\alpha \cdot \mathbf{c}_n{}^{\mathrm{H}} \cdot \mathrm{sgn}(e_j)\right\| = \sum_{j=0}^{M-1} \alpha \cdot \left\|\mathbf{c}_n{}^{\mathrm{H}}\right\| = M\alpha\sqrt{N}\,. \tag{8}$$

From (7) with the assumption that $\hat{\mathbf{x}}_M \approx \mathbf{x}$ (the observer is in steady state at time instant M) the estimation of the settling time is:

$$M \ge \frac{\|\mathbf{x}\|}{\alpha\sqrt{N}}\,. \tag{9}$$

(6) and (9) pose contradictory conditions for the observer. The following section introduces the improved version of the observer which ensures fairly fast convergence with small steady state error.

IV. The Improved Sign Error Observer Structure

In order to resolve the above mentioned contradictory conditions an adaptive tuning of the convergence parameter is proposed:

$$\beta = \alpha v = \alpha \|\mathbf{e}_m\|_1\,; \quad \mathbf{e}_m = [e_m\, e_{m\text{-}1}\, \ldots\, e_{m\text{-}V+1}]^{\mathrm{T}}, \tag{10}$$

where β is the new convergence parameter. $v = \|\mathbf{e}_m\|_1$, $\mathbf{e}_m$ is a vector consisting of the last V values of the error signal at the time instant m when β is modified. $\|\cdot\|_1$ denotes the absolute value norm. It can be called normalized sign error spectral observer. The updating algorithm is the following:

$$\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{x}}_n + \alpha \mathbf{c}_n{}^{\mathrm{H}} \cdot \|\mathbf{e}_m\|_1 \cdot \mathrm{sgn}(e_n)\,; \quad \mathbf{g}_n = \alpha \|\mathbf{e}_m\|_1 \mathbf{c}_n{}^{\mathrm{H}}. \tag{11}$$

If the value of the error signal is high then v is also high, so the state variables are updated more radically (with larger steps), thus the convergence is faster. If the estimation error is low—the estimated and real value of $\mathbf{x}$ are near to each other— $\hat{\mathbf{x}}_n$ is updated with lower modifications so decreasing the error of the observation. These facts mean that the utilization of the norm of the error improves the behavior of the sign error observer. The frequently the parameter v is calculated the faster the convergence is. If $V = 1$, the original observer is obtained.

The optimal value of α in (10) and (11) can be calculated for the case when resonators are aligned uniformly and β is updated in each period of y_n. Let's denote the k-th period of the signal by k. These conditions mean that $V=N$, and $m=kN$ in (10), so $\mathbf{e}_m=\mathbf{e}_{kN}=[e_{kN}\ldots e_{kN-N+1}]$, since for uniformly aligned resonators the length of one period of the signal is N.

In these circumstances the observer algorithm minimizes $\|\mathbf{e}_m\|^2$, thus it makes the power (i.e. mean square) of one period of the error signal minimal if the optimal α is utilized:

$$\alpha_{\mathrm{opt}} = \frac{1}{N \cdot N_{\mathrm{NZ}}}\,, \tag{12}$$

where N_{NZ} is the number of nonzero elements of $\mathbf{e}_m$. In practice this result can be used as an initial value when the refreshing of the convergence factor is taken place with other period or resonators are placed unevenly.

For this structure the convergence of the algorithm depends on the properties of the signal. Let assume that α_{opt} is used. N step convergence can be achieved if all elements of the error signal $\mathbf{e}_{kN}$ in (10) have the same absolute value: $|e_i|=|e_j|: \forall i, j \in [kN...kN-N+1]$. In worst case the error signal is a periodic impulse: except of one dominant element of the period that is $e_I=A$, the other elements are nearly zero: $e_i \to 0$, but $|e_i|>0$ that is important in the calculation of $\mathrm{sgn}(e_i)$. In this case the ratio of the mean square values of consecutive error periods is: $\lambda = \left\|\mathbf{e}_{(k+1)N}\right\|^2 \Big/ \left\|\mathbf{e}_{kN}\right\|^2 = \left(1 - \frac{1}{N}\right)$. Using this worst case value of the decreasing ratio a higher bound for the settling time can be given. Let M denote the number of periods during which the power of error decreases to its ρ-th part. Using these conditions:

$$M \le -\frac{\lg(\rho)}{\lg(\lambda)}\,. \tag{13}$$

V. Results

The preliminary practical results with the introduced sign error spectral observer were achieved in a resonator based wireless active noise control (ANC) system [6][7]. ANC systems are special kind of control systems, where the plant to be controlled is an acoustic one [8]. The controller algorithm is a variant of the spectral observer [7], where the error signal e_n is the noise that is sensed by a microphone.

In our system the noise is sensed by a wireless sensor that samples the error signal (i.e. remaining noise), performs the calculation of the signum function and the norm of the error signal and sends the data to a DSP that implements the observer structure. The sampling frequency of the error signal is 1.8 kHz. Due to the utilization of the normalized sign error observer the amount of the data to be transmitted from the sensor to the DSP was one sixth than that in the case of normal observer. The reason is that instead of the current value of the signal only the sign of the error and the absolute norm of the error in V=32 samples long intervals were transmitted. The data reduction is important in this system because the bandwidth of the communication channel (250 kbps) is relatively low compared to the sampling frequency (some kilohertz). This kind of signal compression makes possible either the expansion of the number of sensors with the same sampling frequency, or the increase of the sampling frequency.

VI. Conclusions and future plans

This paper introduced a simple sign error and a normalized sign error spectral observer that can be deployed in systems with limited resources. These spectral observer algorithms are advantageous because they require reduced amount of information for the updating of the state variables, since the sign of the error signal can be represented by lower number of bits than the value of the error signal. The computational requirements can also be reduced since the multiplication with the sign of the error instead of the value of the error can be substituted by a simple addition or subtraction.

In the paper the transient and steady state properties of the algorithms were also derived. It was shown that the simple sign error observer requires tradeoff between the accuracy and speed of adaptation. In order to relieve these contradictory conditions a normalized sign error spectral observer was presented. This algorithm utilizes the first norm of the error signal for tuning the parameters of adaptation, and provides faster convergence with small steady state error. As a practical application a wireless active noise control system was introduced.

The aim of the future work is the extension of the structure on MIMO case, as well.

References

[1] Widrow, B., S.D. Stearns, “Adaptive Signal Processing,” Prentice Hall, Inc. 1985.

[2] Péceli, G., “A common structure for recursive discrete transforms,” IEEE Trans. Circuits Syst. Vol. CAS-33, pp.1035-1036, Oct. 1986.

[3] Widrow, B., P. Baudrenghien, M. Vetterli, P. Titchener, “Fundamental relations between the LMS algorithm and the DFT,” *IEEE Trans. on Circuits and Syst.*, Vol. 34, Jul. 1987. pp. 814-820

[4] Gersho, A., “Adaptive filtering with binary reinforcement,” IEEE Transactions on Information Theory, Vol. 30, Mar. 1984, pp. 191-199

[5] Bucklew, J.A., Kurtz, T.G., Sethares, W.A., “Weak convergence and local stability properties of fixed step size recursive algorithms,” *IEEE Transactions on Information Theory*, Vol. 39, May 1993 pp. 966 - 978

[6] Orosz, Gy., L. Sujbert, G. Péceli, “Testbed for Wireless Adaptive Signal Processing Systems,” Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Warsaw, Poland, May 1-3, 2007

[7] Sujbert, L., G. Péceli, “Periodic noise cancellation using resonator based controller,” 1997 Int. Symp. on Active Control of Sound and Vibration, ACTIVE ’97, pp. 905-916, Budapest, Hungary, Aug. 1997.

[8] Kuo, S. M., D. R. Morgan, “Active Noise Control: A Tutorial Review,” in Proceedings of the IEEE, vol. 87. No. 6., pp. 943-973, June. 1999.

SCALABLE MODEL CHECKING OF QUORUM CONSENSUS PROTOCOLS

Péter BOKOR
Advisor: András PATARICZA

I. Informal Approach: A Roadmap

Formal verification of *distributed fault-tolerant (FT) protocols* is a non-trivial task. Due to their relative complexity automatic analysis approaches are preferred to avoid human interference. A widely used technique is *model checking* [1] which contains simulating every possible execution of the system to guarantee completeness. Unfortunately, model checking suffers from *state space explosion* when the size of the verification space grows rapidly (even exponential) and an exhaustive exploration is no longer viable. Various techniques exist to mitigate this complexity issue. Some techniques like symbolic model checking propose a general solution. Another main course is often referred to as *abstraction* where the symmetry of the system is used to reduce the verification complexity (e.g., symmetry reduction [2]). An emerging approach related to general abstractions is called *semantic abstraction* which exploits symmetries of a specific class of applications to achieve a considerable amount of complexity reduction [3].

This paper presents a semantic abstraction to support model checking of a broad class of *consensus protocols*. The main assumption about the class of systems (being a semantic approach) restricts the applied "convergence" (or *majority*) function of the protocol. Other characteristics like synchrony and fault model are not constrained. Majority is used to compare the proposed values of other nodes to decide whether the value can be decided or a new value is proposed (and which one).

The main idea of the abstraction is to computing majority based on *symbolic* constants representing the real values of the protocol. Our main premise is that the protocol under examination defines *consensus properties* where no query distinguishes between data values (only their equivalence matters). For example, assume a correct node receiving 1, 1, 1 and 1 from other correct nodes and ?,? from asymmetric value-faulty nodes. Even in the binary case, the model checker needs to distinguish 4 different scenarios per node, therefore, the overall (global) number of scenarios is exponential in the number of nodes. However, each node will decide $Majority(1, 1, 1, 1, ?, ?) = 1$ as enough redundancy is present to tolerate faults. A symbolic representation of the (global) traffic of messages suffices computing $Majority(A, A, A, A, B, C) = A$ which covers all real (i.e., non-abstract) configurations. In this model, A, B and C are symbolic constants.

Note that the previous computational model is not always viable. For example, a function which returns the mean of the input values cannot be computed via symbolic constants. Therefore, we restrict to functions that count the appearances of the input values and return a value v defined a the *quorum* (we say that "v is in majority"). For example, if the required quorum is $1/2n$ (n denotes the number of nodes), the output is the simple majority among the input values. If no v is in majority, a constant `NO_MAJ` is returned or one of the input values is chosen. For the later case, consider the case when the smallest input value shall be returned if not enough equal values are received. The verification of such protocols cannot directly use the symbolic approach. The output of $Majority(A, B, C, D)$, for example, cannot be determined without knowing the actual values even with an ordering relation across the symbols. Considering a natural ordering among English letters, $Majority(A, B, C, D) = A$ and $Majority(A, B, C, E) = A$ which could misleadingly prove consensus if E represents the smallest value among all.

The *benefit* of using the symbolic abstraction (if possible) is the drastic reduction of data-dependent

branches compared to the protocol's real execution. We propose a *hybrid simulation* when the symbolic execution is an over-approximation. In such cases, the data-rich model is used to find the possible value assignments of the symbols such that the quorum function can correctly be evaluated. The symbolic simulation can then continue. In the previous example, $Majority(A, B, C, D)$ can safely return A if the assignments $A \in \{1\}$ and $B, C, D \in \{2, 3, 4\}$ are known from the non-abstract model. In the favorable case where the abstract model suffices and the hybrid approach is not required, our experiments report that model checking is able to verify the abstract model of an example protocol with up to three faulty nodes just within 1 minute, whereas the non-abstract case yielded state space explosion using the same average desktop computer.

II. A Formal Language to Specify Quorum-based Consensus Protocols

In this section, we define the formal basis of the proposed abstraction. In fact, the system model together with the notion of faults are described which can be then used to prepare an abstraction of these basic constructs.

A. Application Logic: Quorum Majority

The following definition describes the core of computation in consensus protocols. Such functions take input parameters from a set D and output a value called the "majority among the input values" from D. Note that, in the informal sense, no majority is required, smaller quorums like $1/3$ of the values are also allowed.

Definition 1 *(*Majority function*)*
$maj^k : D^k \rightarrow D$ *is a* majority function *defined for* $k = 1, 2, \ldots$ *as follows. If k=1 then* $maj^1(d) = d$ *for all* $d \in D$. *Otherwise* $maj^k(d_1, \ldots, d_k) = d$ *if a* quorum $Q = \{d_{e_1}, \ldots, d_{e_l}\}$ *exists (*$l \leq k$*), where* $d_{e_j} = d$ *for all* j. *If no such* Q *exists (or* d *is not unique) a deterministic value from* D *or a pre-defined constant* `NO_MAJ` *is returned. In the former case* maj^k *is called* value-driven *(otherwise* value-dependent*).*

A generalization of the previous definition could be to allow "embedded" (or "hierarchical") application of majority functions. For example, majority can first be computed based on a set of vectors (a majority value for each vector is returned) which is then followed by a voting among the returned values (see Byzantine Agreement [4] as an example protocol for such voting functions).

B. The System Model

The system comprises n *nodes*, each of them executing the same consensus protocol. The program of the protocol maintains a set of *state variables* $V = \{v_1, v_2, ...\}$. The set of all variables is denoted by $\vec{V}$, where $v_j^k \in \vec{V}$ denotes v_j in node k. The domain of each variable is D. Initially, each variable takes a value from $D^I \subseteq D$. This usually corresponds to the *proposed value* of the node.

The execution model is governed by *rounds* following the synchronous lock-step scheme as presented in [5]: (Step 1) At round i, every node *broadcasts* its message, (Step 2) every node collects (delivers) the messages and updates its state variables via the *state transition function* which is restricted to be a set of majority functions. Note that this scheme does not restrict to synchronous systems (see Section D), Step 1-2 only assume an abstract coordination (without the details about its implementation) among the distributed nodes. Furthermore, we remark that the assumptions about the same program in every node and the broadcast communication can be both relaxed. We only use these special (and still general) cases for the simplicity of the discussion.

C. Properties

Our focus is on the usual properties of consensus protocols. Such properties define the equivalence (or non-equivalence) of certain values. We allow using temporal operators to define requirements

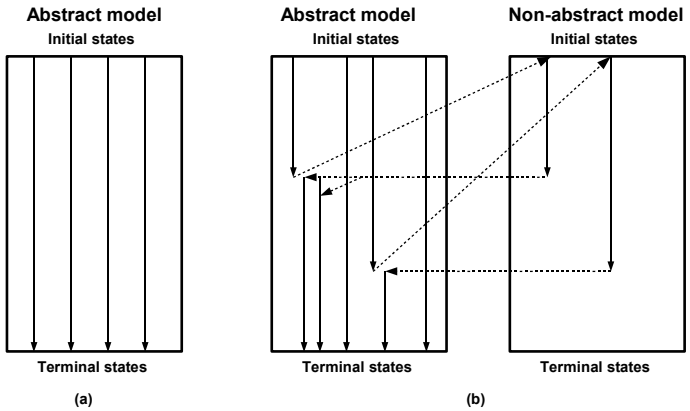


Figure 1: (a) Value-independent protocols (b) Value-driven protocols

across rounds. Informally, atomic properties are evaluated based on the current values of state variables and the further evaluation is done via the standard semantics of temporal logics. For example, the following consensus property defines "validity" in a two-round consensus: $\varphi = \forall j, l : \texttt{proposed}^j = \texttt{proposed}^l \Rightarrow \mathsf{XX}\ \forall k : \texttt{adopted}^k = \texttt{proposed}^l$. φ requires that every node shall adopt the value (and no other one) initially proposed by all nodes.

Formally, atomic *consensus properties* are constructed over $\overrightarrow{V}$ by using the equivalence relation "=" and the standard Boolean operators. Atomic properties can be combined via temporal logic operators.

D. The Fault Model

Faults are distinguished whether they are *benign* (e.g., crash) or *value faults*. The former type can be detected locally being (a) an improperly formatted message (denoted by `ERR` at receipt) or (b) missing message (`MISS`). A missing message is due to crash faults or communication time-out (asynchronous system). Value faults mean messages sent from the domain $D' \subseteq D$ (and delivered on time), however, the message differs from the specified value. By definition, such faults cannot be locally detected as erroneous. Byzantine (*malicious*) faults are a special case where $D' = D$.

In out treatment, similar to [3], the possible faults and asynchrony of the system is captured by the *delivery function*. Formally, the function $rec^k(d)$ returns the set of messages possibly received by a correct node if node k is the sender. For example, $rec^1(d) = \{MISS\}$ if the message sent by node 1 cannot be delivered on an asynchronous link.

III. The Proposed Model Checking of Consensus Protocols

The abstraction base. The main idea of our abstraction is the symbolic representation of the values of state variables. We use a set of symbols $Symb = \{A, B, \ldots\}$ for this purpose. The equivalence relation "=" is naturally defined over $Symb$, for example, $A = A$ but $A \neq B$. Say that $d \in D$ is abstracted by A. The message received by a correct node is abstracted by A if and only if d was delivered on time and the sender was correct. Otherwise, a new symbol is used to abstract the corrupted (or lost) value. The rationale of this abstraction is that consensus protocols usually assume that correctly delivered messages are in majority and erroneous senders are compensated in the majority voting. The formalization of the abstraction (by giving the abstract counterparts of maj^k and rec^k) is omitted here due to the lack of space.

Automated verification. We now outline the proposed verification framework for consensus protocols. Our focus is on automated verification requiring only basic skills in mathematics. We propose using the abstract model of the protocol, instead of the non-abstract one, if it is possible (Figure 1). The approach is "best-effort" in the sense that it might not benefit from the abstraction at

r	Agreement		Validity	
	Non-abstract	**Abstract**	**Non-abstract**	**Abstract**
1	0.38s	0.37s	0.44s	0.39s
2	2.17s	1.03s	2.17s	1.03s
3	–	1m	–	58s

Table 1: Model checking Byzantine Agreement with up to 2 faults using `sal-bmc`

all under unfavorable circumstances. Every time the symbolic execution fails the verification evaluates the possible assignments of the symbols by using the *non-abstract* model. Each possible assignment, corresponding to valid protocol runs based on the assumption the the non-abstract model is a correct one, determines the (correct) symbolic output of the majority function. For example, if $Majority(A, B, C, D) =?$ and the non-abstract simulation determines $A, B \in \{3\}$ and $C, D \in \{1, 2\}$ (e.g., C and D were sent by asymmetric faulty nodes), the symbolic verification continues along two paths having $Majority(A, B, C, D) = C$ and $Majority(A, B, C, D) = D$, respectively. We remark that several optimizations can be applied to circumvent the need of building a full-fledged model of the non-abstract protocol. Once again, due to the lack of space, details are not given here.

Experiments. We run initial experiments to test our abstraction framework. We used the example of a synchronous, value-dependent protocol (Byzantine Agreement [4]) and an asynchronous, value-driven protocol (OneThirdRule [6]) to verify its usual consensus properties, agreement and validity (and liveness in the asynchronous case). The bounded model-checker of the Symbolic Analysis Laboratory (SAL) turned out to be applicable in both cases yielding an effective model checking. The results of verifying the first protocol while raising the number of faults (and thus the number of rounds) are depicted in Table 1. It can be seen that model checking scales for $n > 3$ only if abstraction was used. Initial results for the second protocol have been obtained where, upon each failure of symbolic evaluation, the non-abstract model was run to termination instead of returning to the abstract execution. This only yielded proportional reduction with respect to the pure non-abstract verification (e.g., 11079 versus 14218 states). We believe, however, that a reduction to the extent of magnitudes can be reached with the approach explained in Figure 1.

Proof of correctness. Although the symbolic abstraction is very intuitive, its formal proof of correctness is required when used in the certification of mission-critical applications. This means to prove that a consensus property is true in the abstract model if and only if it is in the non-abstract one. Therefore, we defined the non-abstract and abstract model of general consensus protocols and showed a bi-simulation between them. This result also implies the property preservation as it was stated above.

References

[1] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*, MIT Press, 2000.

[2] E. M. Clarke *et al.* "Exploiting symmetry in temporal logic model checking," in *Proc. CAV*, pp. 450–462, 1993.

[3] P. Bokor *et al.* "Model Checking of Distributed Depandeble Protocols Using Semantic Property Preserving Abstractions," TR, TU Darmstadt, 2007.

[4] L. Lamport *et al.* "The Byzantine Generals Problem," *ACM Trans. on Prog. Lang. and Sys.*, 4(3), 1982.

[5] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann, 1996.

[6] T. Tsuchiya and A. Schiper, "Model checking of consensus algorithm," in *Proc. SRDS*, 2007.

MODEL DRIVEN DESIGN FOR STRUCTURAL RECONFIGURATION BASED DEPENDABILITY

Imre KOCSIS
Advisor: András PATARICZA

I. Introduction

Large, distributed IT infrastructures providing business-critical services have to protect themselves against internal and external threats and adapt to changing environmental parameters, as load. Most widely applied, structural resilience mechanisms are structure-preserving in the sense that the structure of the system does not change as a reaction. However, recently both the growth in size of distributed systems and technological advancements in various fields as networking technologies and virtualization have been acting as enablers for on-line structural reconfiguration.

This paper introduces the generic concept of structural reconfiguration, explores its modeling and design aspects and outlines a model based, integrated configuration and reconfiguration design workflow for virtualized data centers. Following that, verification aspects of reconfiguration based IT resilience are reviewed and some initial results are described.

II. Structural Reconfiguration Based Resilience of IT Systems

Structural reconfiguration is reallocation of computation, storage or communication that is not only a local, parametric reconfiguration of a system component, but one that changes the service delivery structure and the dependencies between the components in runtime. Industrial practice employs numerous standard mechanisms that are structural reconfiguration approaches, as for example server node failover to hot/cold spares in case of physical or software failure, dynamic reallocation of tasks accompanied with software redistribution in data centers upon changing load and rerouting of communication paths based on changing load or connection failures.

While these and other approaches are common practice, research for the systematic handling of structural reconfiguration based resilience of IT systems has begun only lately with the widespread adoption of the key enabling technologies.

A. Structural Configuration and Structural Reconfigurations

Capturing IT system component and system configuration is a domain-specific modeling task: configuration concepts vary largely among IT management domains. Although standards and quasi-standards exist for configuration modeling, most notably the management data metamodels of system management standards and the vendor-specific metamodels of configuration management databases (see e.g. [1] and [2]), even their typical usage is to utilize only a restricted metamodel of interest with custom extensions.

The *structural configuration* of an IT system is the set of relationships between its components as data flows, resource usage or data carrier relationships, server/client relationships, and so on. Meanwhile, individual components also have configurations; however, they constitute the *parametric configuration* part of the system configuration.

B. Diagnostics and Structural Reconfiguration

Structural reconfiguration by its nature depends on a *diagnostic image*, built from the measurements of the sensors deployed in the system. Diagnostic models can be ones that describe the knowledge of faults, errors and failures in the classic sense; however, the term is used here in an extended sense, as

reconfiguration can also be governed e.g. by a performance or a security model. The simplest type of diagnostic images is a vector of state attributes defined on a subset of the immutable components of the system; that is, components whose state is the diagnostic information are not added or removed. Binary diagnosis of hardware components (OK/FAILED) is generally of this type. The diagnostic image influences reconfiguration in two ways.

First, it can be an initiator: the current knowledge of the dynamic properties of the system is what drives the decisions whether a reconfiguration is needed and what are the problems in the structure that have to be dealt with, as failed, compromised or overloaded components. Second, it is a source of constraints on the target configuration; the goal of reconfiguration is to eliminate or avoid the effect of the known issues, identified by the diagnostic image. Thus, these are not to be reproduced; e.g. the target configuration must not allocate services to a physically failed server.

The need for a diagnostic image makes maximum speed and minimal-invasiveness of reconfiguration contradicting requirements, as fine granular diagnosis takes more time than rough, high-level ones.

C. Modeling Reconfiguration in the Engineering Domain

Configuration metamodels usually can be extended to incorporate diagnostic attributes and relationships (for example fault/error/failure modes, error propagation paths). A reconfiguration rule is the IT management workflow to be enforced under certain diagnostic images, leading from one configuration to another. Accordingly, reconfiguration can be described *by the target configuration* (declaratively) or *as a process* (imperatively), both in terms of the pre-established extended configuration metamodel.

Reconfiguration description by source and target configuration pairs, although needs additional workflow synthesis, offers the advantage of capturing diagnostic images in the same model. An aspect of declarative reconfiguration description currently under research is that a partial source configuration equipped with constraints is effectively a left hand side of a graph transformation rule on the configuration metamodel. This way, diagnostic image based reconfiguration initiation can be implemented as an incremental graph pattern search.

D. Structural Reconfiguration and Supervisory Systems

Traditionally, system management is realized in a strictly centralized manner with decisively human beings carrying out diagnosis and deciding on the actions to be taken. However, the IBM Autonomic Computing Initiative [3] and other efforts have been calling for 'closing the loop' with the automation of diagnosis and action planning since the beginning of the decade.

In structurally static IT architectures, design for Quality of Service takes system management into account mostly as a reactive repair process, acting against fault processes on a per component basis. Thus, system design and supervision/support design can be decoupled in an assume-guarantee manner via probabilistic component fault/failure and repair properties. In contrast, in a system where (semi-)automatic structural reconfiguration is a resilience mechanism, overall Quality of Service (QoS) is a derivative of the QoS properties of multiple configurations and the QoS shown during reconfiguration. Therefore, design for user-observable service quality must take not only the QoS of individual configurations, but also *Quality of Management (QoM)* aspects as e.g. switchover times or time to reconfiguration decision into account. These needs call for an integrated, model-based design process for environments managed with structural reconfiguration.

E. An Experimental Design Workflow for Virtualized Data Centers

Currently no unifying model driven design process template exists for the integrated design of configurations and reconfigurations. With the author as project co-supervisor, an experimental design workflow for a constrained problem was formulated and prototyped in [4]. This work focused on virtualized data centers with a fixed set of host machines, a set of virtual machine images and additional performance and dependability constraints, e.g. the minimal number of replicas to be provided for each image to express cluster size and standby needs.

A configuration metamodel – deployment of independent images on physical hosts with piecewise linear modeling of the host resource allowance - performance relations – and diagnostic metamodel (physical host failures) were formulated; models of these are transformed to a a mixed integer linear programming (MILP) problem for finding alternative, repairing configurations based on an initial configuration and its diagnostic image. Ilog CPLEX was used as an industry-strength MILP solver [5]. The user-defined optimization objectives may cover an arbitrary mix of goals like maximization of the availability of high priority services or minimizing the operational costs originating from resource usage. IBM Tivoli Provisioning Manager was used as an IT workflow automation platform. The resulting workflow is depicted on Figure 1. Currently, the logic of the workflow synthesis from the declarative solver results is quite ad-hoc and rudimentary; we are currently examining the possibilities for synthesizing the workflow during problem solving, as well.

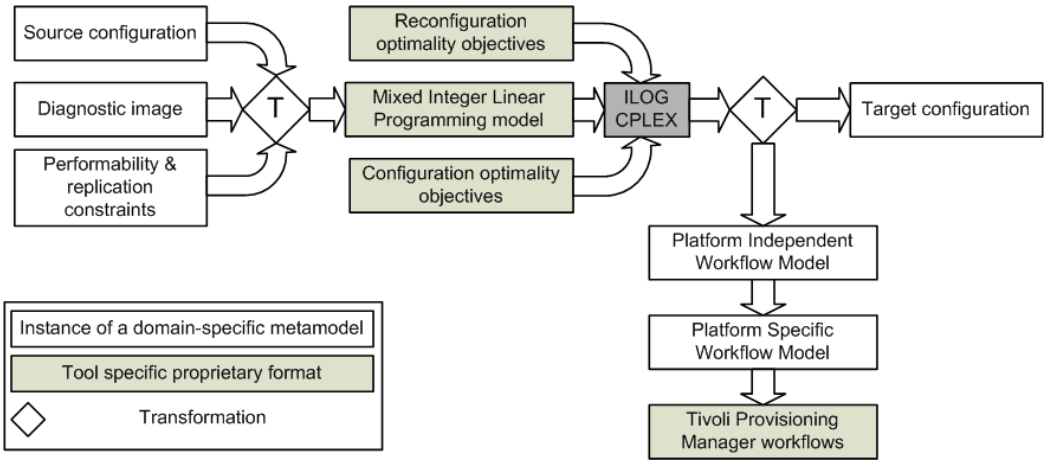


Figure 1: Overview of the experimental configuration and reconfiguration design workflow

III. Dependability and Quality Aspects of Reconfiguration

Model driven design enables for automatic verification of system properties of interest via hidden formal methods. For static infrastructures, service and system related metrics of various fields as performance, availability or safety engineering are well known. However, systems employing structural reconfiguration impose new challenges in terms of properties to be verified.

- Although the definition of service level properties remains the same, analytic modeling and mathematical solving needs to be modified to take into account the supervision-initiated reconfigurations.
- The system is in a transient state during reconfiguration. System dependability and performance can be relaxed during reconfiguration only to the level explicitly accepted as tolerable.
- Measures are needed to be able to express the 'operational range' of the system under the known influencing conditions, as they are used for structurally static dependability mechanisms (for example, number of faults tolerated).

In the following, some initial results are reported.

A. Uninterpreted Modeling and Analysis of Reconfiguration

Uninterpreted modeling of reconfiguration is modeling by using the domain-invariant concepts of the mechanism: configurations, reconfiguration workflows and diagnostic image initiated reconfigurations.

A system utilizing structural reconfiguration can be treated as a state machine, with configurations as states, reconfigurations as transitions and diagnostic images as guard conditions. In the following, we suppose that the components under diagnosis are immutable by the reconfiguration and that the

diagnostic vector is the same for all configurations. Among others, this formalism allows to check the completeness and the determinacy of reconfiguration rule sets.

Let us construct another, probabilistic state machine, where the state vector is the fault and failure state vector of the components modeled. Then let us extend this state vector with the diagnostic image, where the diagnostic infrastructure and diagnostic logic can be pessimistically approximated by deterministic, timed transitions. Also, let us extend the reconfiguration model with transition times. Let us derive the product automaton of these two automata, with its state vector extended with the (pessimistic) service failure modes that is a function of the actual dependability state and the current configuration. This automaton represents the uninterpreted managed behavior of the system in terms of faults and failures. This state machine is amenable to probabilistic analysis as well as model checking of the user observable service and the management logic.

B. Reconfigurability as a Quality of Management Measure

Intuitively, the 'more reconfigurable' a system is, the more possibilities system management has to handle incidents and environmental changes. Reconfigurability measures may aim at capturing:

- the number of configurations reachable from a given configuration on a per trigger event type basis (*reconfigurability of configurations*),
- process characteristics as for example execution time (*quality of reconfiguration*), or
- the (minimal/maximum/average) length of the reconfiguration series path from an initial configuration that leads to a configuration that has no reconfigurability directive for a condition that needs handling (*reconfiguration run lengths*).

Additionally, reconfigurability measures have three fundamentally different contexts.

- *Inherent reconfigurability measures* are those that an idealistic supervisory system would realize on a given environment. These act as upper bounds for implemented solutions and are defined by the available resources and the needs of the services to be implemented.
- *Supervision design reconfigurability measures* are those that characterize the configuration state space and its transitions determined in system and supervision design time. In practical cases, this configuration transition system is likely to be a small subset of all possible configurations and reconfigurations; however, to fully utilize the resilience possibilities inherent to the system, reconfiguration run lengths for typical trigger event sequences and quality of reconfiguration measures should be comparable with the inherent ones (provided that those can be estimated at least).
- *Managed reconfigurability measures* are interpreted on the reconfigurations enabled by the current internal diagnostic image of a supervisory system at any given point of time. Here, the main issue is that how well and how fast supervision converges to the designed measures.

Acknowledgement

The work reported on in the paper was partially supported by the European FP6 project DESEREC.

References

[1] W. Bumpus et al., *Common Information Model: Implementing the Object Model for Enterprise Management*, John Wiley & Sons, 2000.

[2] H. Madduri, S. Shi, R. Baker, N. Ayachitula, L. Shwartz, M. Surendra, C. Corley, M. Benantar, and S. Patel, "A configuration management database architecture in support of IBM Service Management," *IBM Systems Journal*, 46(3), 2007.

[3] A. Ganek and T. Corbi, "The dawning of the autonomic computing era," *IBM Systems Journal*, 42(1):5–18, 2003.

[4] I. Váncsa, "Construction of a virtualization-based high-availability server," Scientific Student Conference Report, BME, 2007.

[5] ILOG, "CPLEX 9.0 User Manual," 2004.

STOCHASTIC DEPENDABILITY EVALUATION OF APPLICATIONS IN MOBILE ENVIRONMENTS

Máté KOVÁCS
Advisor: István MAJZIK

I. Introduction

As information technology advances car, ship, train and airplane manufacturers increasingly tend to take advantage of its services. Embedded systems are used to make the vehicles' existing functionalities more reliable and precise, as well as extend the driver's safety and comfort.

The HIDENETS project envisions a traffic environment where vehicles are in constant connection with each other and with the so-called fixed infrastructure by means of ad-hoc wireless network technologies, and use various applications during the travel. The goal of the project is to establish methodologies for the development of highly dependable services and applications based on the unreliable network infrastructure. Examples to these applications are [1]:

- In case of a car accident, it is very important to study the circumstances of the accident to find the right cause. The *distributed black box* application constantly conveys information on position, speed, acceleration, etc. to the surrounding vehicles and the fixed infrastructure, so it can support the investigation of an accident.
- The *platooning* application lets a platoon of cars be driven by only one driver in the front. The rest of the cars autonomously follow the track of the first, meanwhile keeping due distance from each other. This functionality is possible if all the cars are in constant connection with each other, and periodically transmit and receive dynamic information on their positions.
- The *traffic sign extension* allows for the dynamic adjustment of speed limits on road sections and the fine tuning of traffic sign periods of junctions based on the traffic load.

Some of the applications themselves, and the underlying middleware consisting of the HIDENETS services are going to be implemented to demonstrate the usability of the development methodology. In order to guarantee the dependability, multiple verification and evaluation phases have to be involved in the development process of these services and applications.

II. Stochastic Evaluation of the Dependability of Mobile Applications

We present a holistic evaluation workflow (illustrated in Figure 1) that integrates several tools and model transformation steps to compute the probability of the successful execution of a series of user activities (user workflow) in a dynamic HIDENETS environment. In the figure models are represented as parallelograms, while model transformation steps are depicted as straight rectangles.

The dynamic HIDENETS environment means that the user relies on functions built upon the services of the fixed infrastructure and ad-hoc domains, and tries to execute the activities in an environment that is characterized by changes: the activities are distributed (include collaboration with other users), the users may move, and the mobility and network traffic influence the availability and quality of the services. The user is considered as the driver of a car equipped with a HIDENETS node, or the administrator of a component in the fixed infrastructure. In the following we refer to the *scenario* as a concept that involves all user-related and environment-related changes.

The inputs of the evaluation workflow are comprehensive and include the outputs of several specific evaluation steps (that were also developed in the HIDENETS project). The main aspects are the description of the user workflows, the structure of the applications used by the users, the dependability

parameters of specific services or resources included in these applications, the traffic and the mobility pattern.

The three sets of input models (views depicted in a dotted frame in Figure 1) of the scenario being evaluated are as follows:

- Each user workflow specifies the user activities in terms of application usage (the mobility aspects like speed and direction are not addressed in the user workflow, they are included in the topology model). There is a user workflow for each participant of the scenario to be evaluated. The workflow is to be provided by a UML based workflow editor.
- Each topology model represents the information on the evolving ad-hoc topology of network connections. There is a topology model for each technology that can be used for communication. This is to be computed by an existing mobility trace generator and network topology generator tool-chain reported in [2] and in [3].
- The application-service dependency models define how the given applications depend on the services, hardware or software components of nodes. These dependencies are to be described by UML architecture diagrams.

The output of the evaluation, that is the computed probability of successful execution of a user activity sequence, can be used to characterize the user workflow in a best case or worst case situation (e.g., whether it is possible to execute the activities at all in case of extreme network conditions), to compare different execution strategies at the user level (e.g., whether it is reasonable to rely on given functions or it is better to use other ones), or to compare different environment options (e.g., compare routes with different traffic and infrastructure conditions and use the one that is characterized by a higher probability of successful execution for a given workflow).

The evaluation workflow follows the approach of *multilevel modeling addressing phases* [2]. Hierarchical multilevel modeling is used by considering communication, architecture, application and user levels, while the multi-phase approach is utilized to model the changes induced by the user activities and the varying environment conditions.

The architecture of the HIDENETS components are captured by hierarchic models of several layers:

- *User level.* This level provides high-level QoS and resilience attributes as perceived by the users. It describes the users' profiles, that is how the users interact with the application and how their requests are mapped to the different components of the application and the supporting architecture.
- *Application level.* This level describes the system behavior from the application point of view. The applications differ in their technical properties, in their mechanisms, in their interfaces, and they can impose different communication and middleware level requirements.
- *Architecture level.* This is the part of the system capturing the behavior of the main hardware and software components that can affect the application-level measures. It describes how the application components and services of the application level are implemented on these resources.
- *Communication level.* It captures the communication aspects of the system. It addresses the link layer (considering several types of networks like WLANs, UMTS and GPRS), the network layer (IP-based) and the transport layer (considering several types of protocols like TCP and UDP).

III. State-space Reduction Techniques

Given the scenario in the form of input models described in Section II. the evaluation workflow can produce the probability of the successful execution of the involved user workflows. In order to keep the state space of models being directly analyzed under control, multiple techniques are used. The basic idea is that those components and submodels the dependability parameters of which are independent of the environment should be evaluated in isolation.

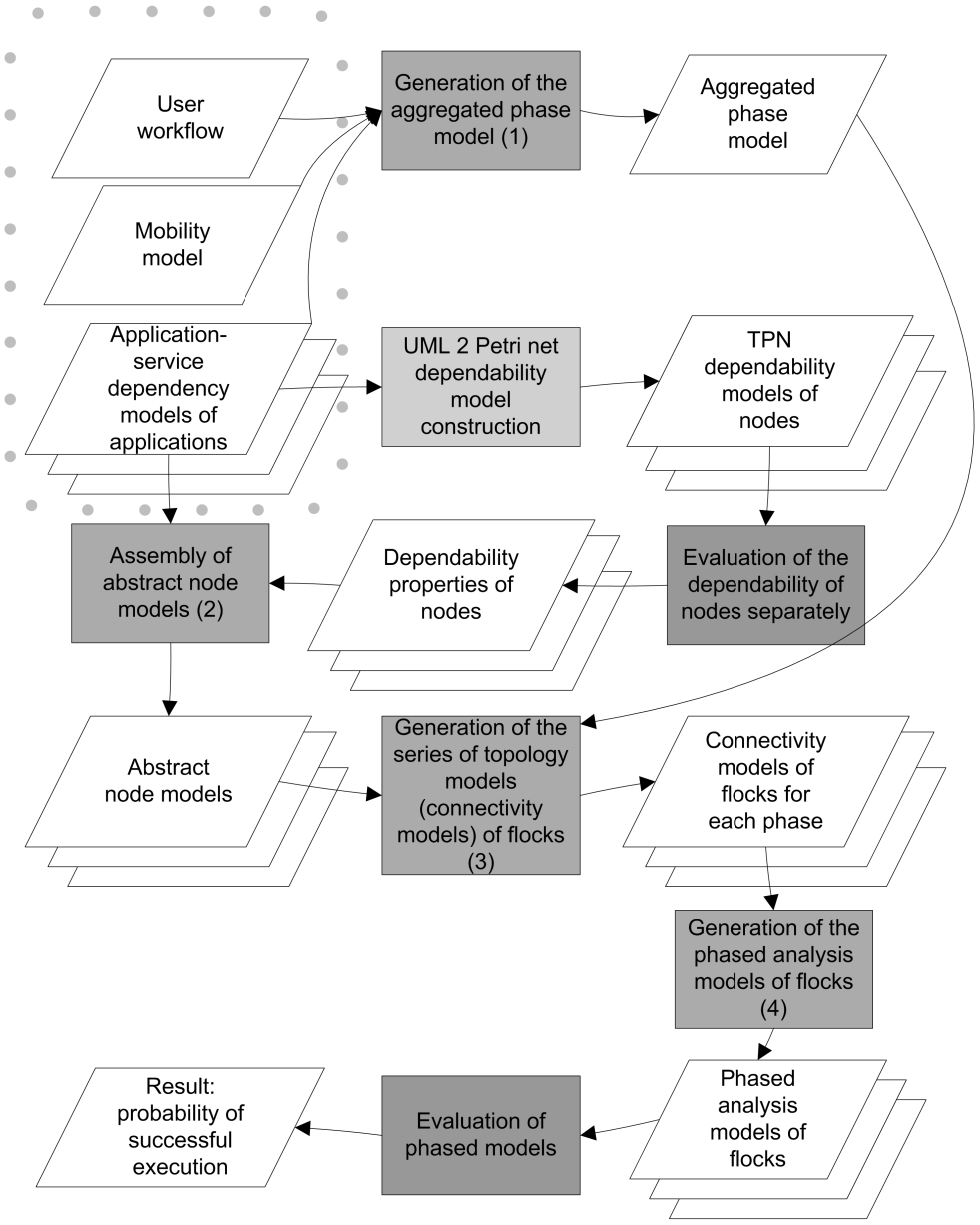


Figure 1: Evaluation Workflow

A. *Evaluating HIDENETS Nodes in Isolation*

The stochastic failure and repair processes of the HIDENETS nodes themselves are independent from that of other nodes and the network connections. As it is illustrated in Figure 1 the dependability models of nodes (generated from object models extended with parameters given in the form of stereotypes and tagged values) are evaluated separately. The applied model transformation [4] produces the Timed Petri Net representation of object models. These models are then solved resulting in the dependability properties of the individual nodes.

Further steps of the evaluation workflow deal with HIDENETS nodes in the form of *abstract nodes*. Abstract nodes represent the composite models of nodes by singular objects with dependability parameters reflecting those of the original ones.

B. *Evaluating Flocks*

The number of nodes participating in a scenario is unlimited. However, the set of nodes in a scenario may have independent subsets in the sense that the members of different sets do not know about the

existence of each other, and do not communicate with each other. These self-contained subsets are called *flocks*. Since flocks are independent, the evaluation of a scenario can be broken down to the evaluation of the flocks found in it.

C. Scenarios as Phased Mission Systems

In this evaluation workflow time is dealt with in a discrete manner. Transformation step 1 in Figure 1 assembles a series of static views that constitutes the scenario. In each view the network topology, the connection parameters and the dependability parameters of nodes are considered static, but any of these might vary during the atomic phase changes. At this stage the scenario is represented as a series of static topology views among abstract nodes. The dependability of these models are evaluated by DEEM [5] or Möbius [6] using an algorithm that only need to keep the state space of one phase in the memory, thus making the evaluation efficient.

IV. Conclusion

The goal of the HIDENETS project is to create dependable services based on unreliable, ad-hoc network infrastructures. Ad-hoc, wireless network connections will provide means to communication for mobile nodes and those of the fixed infrastructure.

The evaluation of such infrastructures and technologies from the dependability point of view is non-trivial for several factors: realistic mobility traces have to be generated, models with large state spaces need to be evaluated, time and dynamicity in the environment have to be dealt with.

This paper introduced a modeling methodology that is capable of capturing every aspect that is necessary to capture a scenario. Additionally we try to tackle the problem of state space explosion with several techniques.

References

[1] M. Radimirsch, E. V. Matthiesen, G. Huszerl, M. Reitenspieß, M. Kaâniche, I. E. Svinnset, A. Casimiro, and L. Falai, "Use case scenarios and preliminary reference model," HIDENETS Deliverable D1.1, October 18 2006.

[2] P. Lollini, A. Bondavalli, J. Arlat, M. Clemetsen, L. Falai, A. F. Hansen, M. B. Hansen, M. Kaâniche, K. Kanoun, M. Kovács, Y. Liu, M. Magyar, I. Majzik, E. V. Matthiesen, A. Nickelsen, J. J. Nielsen, J. G. Rasmussen, T. Renier, and H.-P. Schwefel, "Evaluation methodologies, techniques and tools (final version)," HIDENETS Deliverable D4.1.2, December 17 2007.

[3] M. N. Jensen and A. Nickelsen, "Evaluation of routing dependability in MANETs using a topology emulator," M.S. thesis, Aalborg University, Aalborg, Denmark, 2007.

[4] I. Majzik, A. Pataricza, and A. Bondavalli, "Stochastic dependability analysis of system architecture based on UML models," in *Architecting Dependable Systems*, R. de Lemos, C. Gacek, and A. Romanovsky, Eds., pp. 219–244, Berlin, 2003. Springer Verlag.

[5] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and I. Mura, "Dependability modeling and evaluation of multiple-phased systems using DEEM," *IEEE Transactions on Reliability*, 53(4):509–522, 2004.

[6] University of Illinois at Urbana-Champaign, *Möbius*, 2006.

SPECIFYING TESTS FOR AD-HOC MOBILE SYSTEMS

Zoltán MICSKEI
Advisors: István MAJZIK (BUTE), Hélène WAESELYNCK (LAAS)

I. Introduction

Mobile ad-hoc networks propose new challenges for software development and verification and validation (V&V) activities. Apart from the issues found in fixed distributed systems, fresh ones are presented in the new environment: high dynamicity or context awareness. New nodes are constantly joining and leaving, the application running on the host has to be aware of these changes. Nodes are moving out of each other's communication range frequently, hence the failure of sending a message is not a rare event any more. The state of an application depends not only on the messages it receives from the others, it should also take into account its context, e.g., its current location coordinates supplied by a GPS unit or other information from the environment. Thus the testing methodology of these systems should take into account these specificities. The current modeling languages for specifying test cases have to be adapted to these requirements.

This paper presents the typical problems of mobile systems through a case study. Section II presents the related work in modeling mobile systems, Section III describes the case study and the results of its analysis, while Section IV illustrates with examples why UML 2.0 Sequence Diagrams need extensions when used for specifying tests for mobile systems.

II. Related work

According to our research, currently there is no standard for modeling mobile systems yet, but in the recent years several approaches have emerged. A number of publications focus on mobile agents from the broad area of mobile systems. An agent is a software component that executes specific tasks on behalf of someone with some autonomy [1]. Mobile Agent Modeling with UML is a UML profile recommended by Belloni and Marcos in [1]. The stereotypes and tagged values of the profile are organized into views that describe the different aspects of the mobile agent. In [4] mobile computing (MC) environments were investigated. Because Objectcharts (which are variants of Harel's Statecharts) turned out to be inadequate to model such environments, an extension called Mobicharts was proposed. The extension contained specific states to model, e.g., the situation when the mobile host is disconnected and mechanisms to express task migration.

Grassi et al. proposed an UML profile to support physical mobility of the computing nodes and the logical mobility of software elements [2]. The behavior of mobility was expressed on so-called mobility manager statecharts. The paper included examples to show how the profile can be applied to describe basic mobile code paradigms (e.g. Code on Demand, Mobile Agent). In [3] a UML extension called Mobile UML was proposed to model mobile systems in global computing. The extensions consisted of (i) a UML profile to express mobility concepts (*location, mobile, mobile location*) and (ii) new diagram types. According to the authors, the problem with UML Sequence diagrams when modeling mobile scenarios is that movement of an entity can be expressed only indirectly by adding a new object box. Thus, to overcome the complexity of this approach, a new diagram type Sequence Diagram for Mobility (SDM) was recommended.

Several approaches have been proposed, that contain many similar elements, however, each of them are specialized for a specific aspect of mobile systems, and no general standard is available at the moment. Moreover, these extensions mainly consider logical mobility or physical mobility from one infrastructure point to the other, and they do not offer a solution to ad-hoc networks.

III. GMP Case Study

This chapter includes the insights gained from the detailed analysis of a Group Membership Protocol (GMP) [5]. We choose this particular application because it is a good example of a non-trivial, mobile-based service. It addresses a very complex problem, i.e., to maintain a consistent membership information in a mobile setting, where besides the challenges raised by traditional distributed systems (e.g. atomicity, asynchronous behavior) problems from the mobile environment (e.g. frequent topology changes, network delays) also arise. The protocol has a specification [5] which contains (i) general properties the protocol should satisfy (e.g., the successor of a group shall be either a proper superset or a proper subset of the group), (ii) textual description and (iii) pseudo-codes describing the important methods. Moreover, the authors created a Java language implementation as part of the LIME [6] open source middleware for mobile applications. The implementation is not just a small example program: it consists of 4 KLOC of Java code, contains 22 Java classes and after all the components are started there are 6 concurrent threads.

The functionality of the protocol is divided into two parts: (i) group discovery manages the discovery and reporting of newly arrived hosts, (ii) group reconfiguration performs the merging and splitting of groups when needed. The protocol uses a centralized approach, every group has one leader. The leader collects the location and discovery information of the hosts. Using this information, it checks the group merging and splitting criterion, and starts a group change operation if necessary. The criterion used is the *safe distance* criterion, i.e., if two nodes are within this distance, the protocol guarantees that, regardless of their moving pattern, they will have enough time to finish their current communication.

The analysis was conducted by (1) reviewing the specification, (2) creating a UML model for the implementation of the protocol, (3) comparing the specification to the implementation, and (4) testing the implementation. The general properties of the protocol were analyzed to determine whether they are testable or not. The description of the protocol was reviewed; worst-case scenarios were investigated to see whether the calculation of safe distance and the atomicity of the group changes are always valid. UML class diagrams were created to model the static structure of the implementation, this helped to check conformance to the specification. Sequence diagrams were drawn for each of the important scenarios, which revealed design failures and possibly invalid scenarios. Finally, simple random testing was carried out on the implementation which found several scenarios violating the properties of the protocol. Several of these scenarios were anticipated previously by the review. In summary, the following main issues were found during the review (the detailed description of the analysis and the results can be found in [7]):

- Some of the general properties of the protocol are incomplete or not testable.
- The English language specification of the protocol is sometimes ambiguous, and the pseudo code definition of the key functions is also not sufficient, the control flow is missing.
- The atomicity of the protocol is not guaranteed in worst-case scenarios.
- The implementation lacks key features that are essential to the correct behavior.

The analysis showed general problems that are relevant for any mobile application dealing with mobility and cooperation of hosts. The same analysis (e.g., testable properties, atomicity of operations, identifying unclear parts in the specification and modeling static and dynamic structure) could be performed for any mobile application. Moreover, the case study highlighted the following general challenges:

- It is not easy to model mobile system instances. Without a suitable notation and modeling methodology serious design defects could be introduced.
- The definition of properties containing spatial and temporal information is a complex task, but the correct formulation is essential to the later verification steps.

IV. Expressing Tests using UML 2.0 Sequence Diagrams

UML 2.0 introduced a major change in the sequence diagrams (SD), many new elements were imported from other scenario languages like Message Sequence Charts (MSC). Several operators were introduced to combine diagram fragments, e.g. *parallel* and *alternate* execution. *Negation* and *assert* operators could be used to specify modalities, *ignore* and *consider* operators can express that the diagram shows only a subset of messages. Using the previously presented GMP case study we investigated how these new elements can be used in testing by trying to specify test cases.

Using the UML 2.0 SD language a test case for a split scenario in the GMP can be expressed like the following. The *goal* of the test case is to check that if a node moves out of safe distance the leader detects it and sends the new group membership to everyone before the node leaves the communication range. This behavior can be expressed with the sequence diagram on Figure 1 showing the messages exchanged during the split operation.

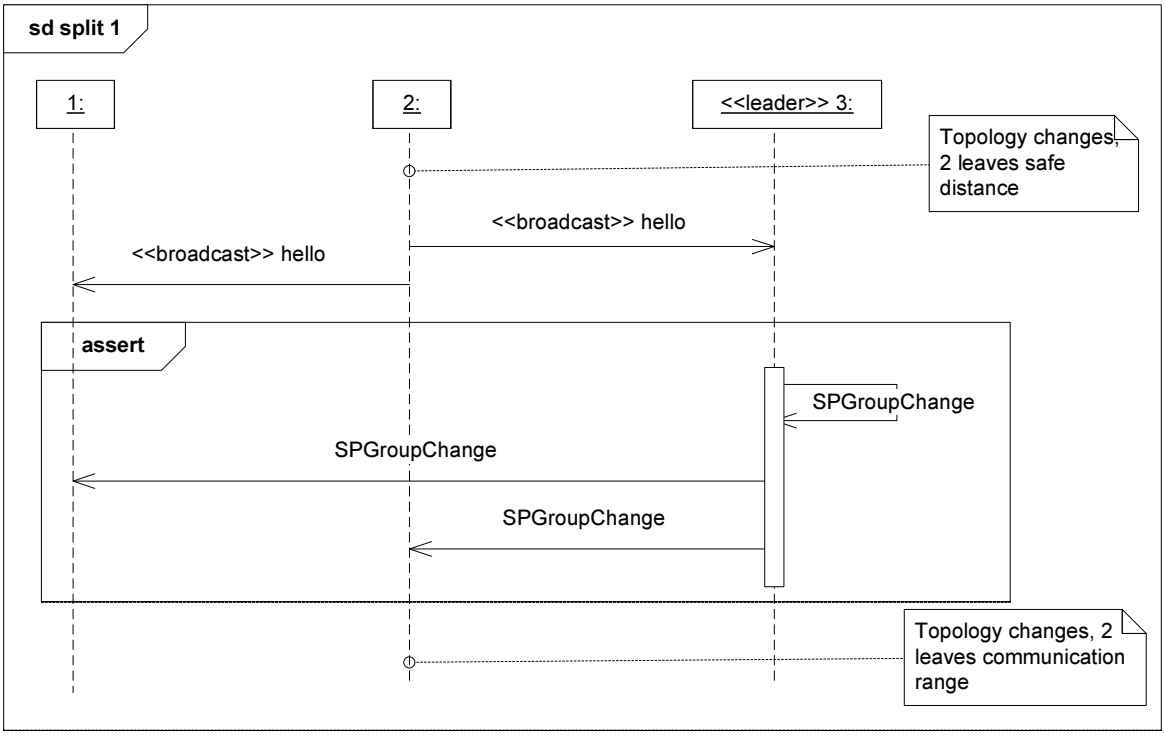


Figure 1: Test scenario for a group split

The above diagram uses only standard UML concepts. Using comments and stereotypes the scenario can be partially described, although it can be seen that UML sequence diagrams lack the element to express two important constructs frequent in mobile environments in a precise way:

- broadcasting messages in local vicinity,
- context changes, like two nodes moving out of each other's range.

Apart from the problem of missing elements to describe the mobile environment, the other challenge with Sequence Diagrams is the lack of well-defined semantics. Semantics problem appear e.g., in the assignment of the verdict. The test is passed if node 1 and node 2 receive the SPGroupChange message with the correct group membership. The *assert* operator could be used to show explicitly what are the required messages, if the messages in the assert fragment do not appear, that particular trace is considered as not valid. However, as reported earlier in literature, e.g., in [8], the definition of assert's semantics is quite problematic.

The next example illustrates a more general semantics issue. Figure 2 is valid according to the UML specification; however it raises serious causal issues. For example sending message *z* must

occur after receiving *y*, because they are on the same lifeline, thus sending *z* is after the sending of *x*. But if *y* is not received (it is contained in an optional fragment), this causal ordering is not valid as sending *x* and sending *z* become concurrent activities.

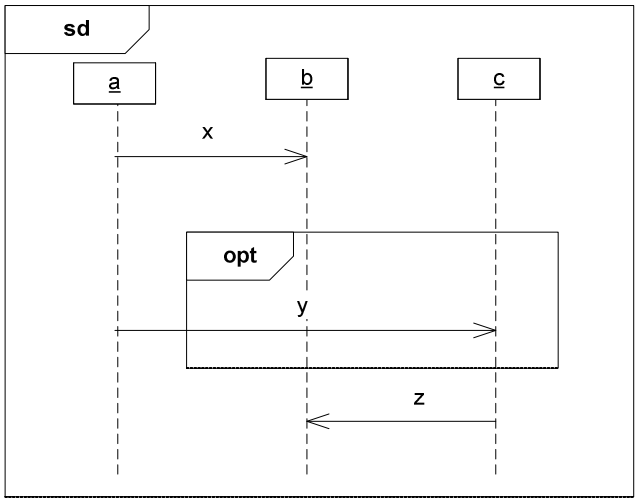


Figure 2: Example for semantic problems in UML Sequence Diagrams

It can be seen even from these examples, that there are several challenges with UML 2.0 Sequence Diagrams if used for test case specification. The UML 2.0 version contains too much complex language elements with ambiguous semantics. Possibly a narrower set of elements, but with a well-defined semantics would be much more useful for specifying precise test cases.

V. Conclusion

In this paper we presented a case study, an analysis of a Group Membership Protocol, which revealed the typical challenges in modeling and testing mobile systems. Moreover, it turned out that UML Sequence Diagrams, which are frequently used for describing test cases, lack the concepts to express the high dynamicity and context awareness of mobile systems. Based on the experiences gained in the case study our future work is to specify extensions, with a well-defined semantics, that adapt Sequence Diagrams to this environment.

References

[1] E. Belloni and C. Marcos, "MAM-UML: An UML Profile for the Modelling of Mobile-Agent Applications," in *Proceedings of the XXIV International Conference of the Chilean Computer Science Society (SCCC'04)*, 2004

[2] V. Grassi, R. Mirandola, and A. Sabetta, "A UML Profile to Model Mobile Systems," in *The Unified Modelling Language, LNCS 3273*, Springer, 2004

[3] H. Baumeister, N. Koch, P. Kosiuczenko, P. Stevens, and M. Wirsing, „UML for Global Computing," In *Proc. of IST/FET Int. Workshop on Global Computing (GC'03)*, Revised Selected Papers, LNCS 2874, Springer, 2003

[4] S. Acharya, R.K. Shyamasundar, "MOBICHARTS: A Notation to Specify Mobile Computing Applications," in *Proc. of 36th Hawaii International Conference on System Sciences (HICSS-36 2003)*, January 6-9, 2003, Big Island, HI, USA. IEEE Computer Society, 2003, ISBN 0-7695-1874-5

[5] Q. Huang, C. Julien, and G. Roman, "Relying on Safe Distance to Achieve Strong Partitionable Group Membership in Ad Hoc Networks," *IEEE Transactions on Mobile Computing* 3, 2 (Apr. 2004).

[6] Lime, Middleware for mobile applications, URL: http://lime.sourceforge.net/

[7] Z. Micskei, H. Waeselynck, M. D. Nguyen, and N. Riviere, "Analysis of a group membership protocol for Ad-hoc networks," LAAS Technical Report *no. 06797*, November 2006

[8] D. Harel and S. Maoz, "Assert and negate revisited: modal semantics for UML sequence diagrams," in *Proc. of SCESM '06*, Shanghai, China, 2006.

MODELING THE PERFORMANCE OF VIRTUALIZATION SYSTEMS

Péter PÁSZTOR
Advisor: András PATARICZA

I. Introduction

Virtualization in general is the abstraction of computing resources. Platform virtualization is a technology, enabling a single computer to act as multiple computers of the same – or very similar – architecture; the subject of this paper is basically this technology. It is an emerging technology in enterprise environments, but its always emphasized advantage, efficiency is double-faced. This paper is part of a work to make a theoretical foundation to be able to approximate, compare, and evaluate the performance capabilities, or bottlenecks of virtualization solutions.

This work explores how the virtualization systems' performance can be modeled. The main aim is to provide a simple, crude method to approximate the needs of a given set of applications in a virtualized environment compared to native hardware environment, and to give a method to compare virtualization methods, which both might serve as the basis of further enhancements.

A. Motivation and historical information

The technology is not new, it was first introduced in the mid 1960s in the IBM M44 system and in 1974, Popek and Goldberg discussed the subject of CPU virtualization [1], and laid down its still valid theoretical foundations, mentioning efficiency as a requirement. However, the recent advances of smaller scale computing made it possible to bring this technology closer to the smaller businesses, or even single users. Several solutions emerged, and now there are tools for every possible requirement – there are free solutions, fault tolerant solutions, universal solutions, efficient solutions, but since some of these are controversial requirements, no solution can be good at all of these at the same time.

The motivation of this work is that to this time, there is no valid approximation of the performance to be expected from a given set of hardware virtualized, and the best way to select the best solution for a given task is almost guessing. It would be very important for such optimization methods, like the one described in [2].

B. Enterprise environment virtualization techniques

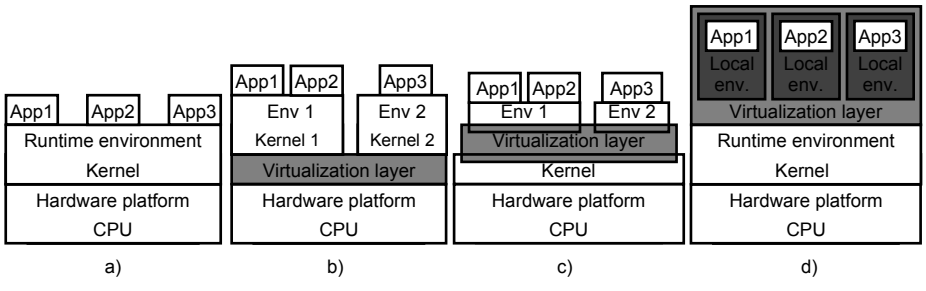


Figure 1: Models of different kinds of virtualization techniques

Figure 1 displays the layers of the architecture in different situations. Part a) is the generic model of a computer running three applications. Part b) displays a generic platform virtualization architecture, part c) is the sketch of operating system level virtualization, and part d) displays application virtualization. This work considers platform and operating system level virtualization.

II. Characterizing performance, workloads and the virtualization system

The performance of a given virtualization solution is a key factor in marketing the products, so the supplied marketing brochures contain a single impressive efficiency value – the best value ever measured, but in the case of a so versatile tool, one number is not enough to characterize it.

A. Performance of the native system

To characterize the maximal performance (P_{max}) of a system, at least the CPU performance, memory, disk and network throughput values are important. These four are performance metrics in the classical way. Their metrics could be: number of CPU instructions processed (P_{cpu}), amount of data transferred to or from the memory (P_{mem}), disk (P_{disk}) and network (P_{net}) per time unit. These values should be measured by running synthetic microbenchmarks of the appropriate field. This measurement has an error, the background load of the operating system, but it is assumed, that this is fairly low, and will serve as a safety margin.

$$\overline{P_{\max}} = \left(P_{cpu} \quad P_{mem} \quad P_{disk} \quad P_{net}\right)^T$$

B. Workload

The next task is to characterize the actual load of the system. This work assumes that the load is a large volume of queries. During normal, average load operation of the system to be virtualized, the utilization of each resource should be monitored, and averaged at a representative period, while also counting the number of queries processed ($n_{queries}$). The utilization of the resources should be also recorded in the idle state of the system ($\overline{M_s}$). From these, an average character of the load can be acquired, and from that, the resource utilization represented by one query can be computed. The utilizations are designated by μ, the utilizations of the background load have S in the index, while the utilizations measured under load have an L in the index, so for example the CPU utilization in the idle state of the system is designated by μ_{Scpu}. The performance loss of the background operations of the system ($\overline{P_{stat}}$) can be formalized by multiplying the utilization vector of the idle system by the maximal performance of the system:

$$\overline{M_s} = \left(\mu_{Scpu} \quad \mu_{Smem} \quad \mu_{Sdisk} \quad \mu_{Snet}\right)^T;\ \overline{P_{stat}} = \overline{M_s}\,\overline{P_{\max}};$$

$$\overline{M_{load}} = \left(\mu_{Lcpu} \quad \mu_{Lmem} \quad \mu_{Ldisk} \quad \mu_{Lnet}\right)^T - \overline{M_s};\ \overline{M_1} = \overline{M_{load}}\,\frac{1}{n_{queries}}$$

C. Performance characterization of the virtualization system

The virtualization system needs to be characterized too. The method is similar to that of the native system, but there are complications. For availability reasons, it is useful to partition the system into more virtual images, meaning more performance losses. The performance of a virtualization system operating in an optimal setup at almost peak load with *n* images can be formalized as:

$$\overline{P_{\max}} = \overline{P_{statovh}} + \sum_{i=1}^{n} \left(\overline{P_i} + \overline{P_{vmovh\_i}}\right) + \overline{P_{idle}}$$

where $\overline{P_{statovh}}$ is the static overhead of the virtualization system, $\overline{P_i}$ is the useful, and $\overline{P_{vmovh\_i}}$ is the useless performance of the virtual machine *i* of *n*, and P_{idle} is the idle performance, the safety margin, which should be at least around $0.3\overline{P_{\max}}$, as illustrated in [3]. The load that can be computed from this equation is the maximal load of the system, which should not be exceeded in average.

The problem is, that $\overline{P_{vmovh\_i}}$ is not trivial to be modeled. A virtualization system performs various kinds of operations in the background by itself, and related to the operations of the different images too. These are very difficult to characterize inductively, not to mention that they might act non-

linearly. This again calls for measurement. But this time, there will be another important factor: the correlation of the utilizations of the separate resources, characterized by this matrix:

$$\overline{\overline{C}} = \begin{pmatrix} 1 & \mu_{mem-cpu} & \mu_{disk-cpu} & \mu_{net-cpu} \\ \mu_{cpu-mem} & 1 & \mu_{disk-mem} & \mu_{net-mem} \\ \mu_{cpu-disk} & \mu_{mem-disk} & 1 & \mu_{net-disk} \\ \mu_{cpu-net} & \mu_{mem-net} & \mu_{disk-net} & 1 \end{pmatrix} \quad (1)$$

Given the complexity of such a system, this matrix is still a very crude model. For example, it does not depend on the number of running virtual images, which would make every μ value a function of n. The values in the matrix in (1) can be determined by running the same microbenchmarks as in the case of the measurements of the hardware, but not only the measured values should be recorded, but the average utilization of all the four resources during the benchmarks. The resource being benchmarked will be fully utilized, hence the ones in the main axis, and the results of the other three resources will show, how they affect each other. After the measurements, the raw data contains error because of the static performance needs, which should be subtracted from the appropriate values, and then the main axis of the matrix should be normalized, so for example the effect of one unit of memory transfer to the CPU utilization ($\mu_{mem-cpu}$) can be computed using the CPU utilization during memory benchmark ($\mu_{membench\_cpu}$) and the CPU utilization in idle state of the system (μ_{stat_cpu}):

$$\mu_{mem-cpu} = \frac{\mu_{membench\_cpu} - \mu_{stat\_cpu}}{1 - \mu_{stat\_cpu}}$$

This matrix is mostly dependant on the virtualization system, and its difference from the native system. Figure 2 explains the probable causes of the values in the matrix, as well as some approximation of which values should be 0.

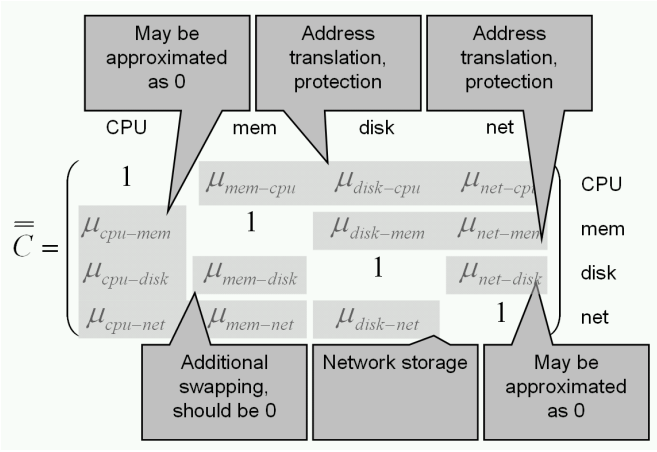


Figure 2: explanation of values in the matrix in (1)

However, the peak performance characteristics are different from that of the hardware system, this can be modeled by using the following matrix ($\overline{\overline{I_4}}$ is the 4x4 unity matrix):

$$\overline{\overline{S}} = \overline{\overline{I_4}} \begin{pmatrix} \frac{P_{cpu\_h}}{P_{cpu\_v}} & \frac{P_{mem\_h}}{P_{mem\_v}} & \frac{P_{disk\_h}}{P_{disk\_v}} & \frac{P_{net\_h}}{P_{net\_v}} \end{pmatrix}^T$$

where P_{cpu_v} is the peak performance of the virtual CPU, P_{cpu_h} is the performance of the hardware CPU and so on. Te result of multiplying $\overline{\overline{C}}$ with $\overline{\overline{S}}$ tells us, how many units of each performance type is consumed by having a load that would consume one unit of performance of each resource.

The total performance used – both useless and useful – by n virtual machines for a load can be computed as:

$$\overline{P_{vm\_total}} = \overline{\overline{CS}}\,\overline{M_{load}} + n\overline{P_{stat\_vm}} = \overline{\overline{CS}}\,\overline{M_1} n_{queries} \overline{P_{\max}} + n\overline{P_{stat\_vm}} \quad (2)$$

where $\overline{P_{stat\_vm}}$ is the performance used by an idling virtual machine, $\overline{M_{load}}$ and $\overline{M_1}$ are defined in II.B. $\overline{\overline{CS}}\,\overline{M_1} n_{queries} \overline{P_{\max}}$ gives the sum of the useful performance of the virtual machines, since the $\overline{\overline{CS}}$ matrix containing the correlation of real-world performance and virtualized performance is multiplied by the vector of the performance used by the load produced by *n* queries ($\overline{M_1} n_{queries} \overline{P_{\max}}$).

D. The result

The next task is to approximate the amount of load the virtualization system can of handle. The target is that the utilization of any resource shouldn't go over 70%, and while providing enough useful performance, availability should be maximized too, by having *n* virtual images. Using (2):

$$\overline{P_{\max}} = \overline{P_{statovh}} + \overline{\overline{CS}}\,\overline{M_1} n_{queries} \overline{P_{\max}} + n\overline{P_{stat\_vm}} + \overline{P_{idle}}$$

now only *n*, $n_{queries}$, and $\overline{P_{idle}}$ are unknown, but given that every components of $\overline{P_{idle}}$ should be at least 30% of the component of $\overline{P_{\max}}$, and given the availability requirements, n_{query} is easy to compute, by selecting the dominant element of the resulting vectors and solving the equation.

E. Problems

On a hardware platform, it can be assumed that in the proper utilization range (below 70% for all resources) the loads act linearly, but in a virtualized environment this cannot be guaranteed, since all the virtual images behave as if they were working on their own separate hardware, while the virtual machine monitor partitions the system under them. Also choosing the appropriate benchmarks, and refining the model is necessary to get more precise approximation, and even then, there might be exceptions from the rules, special cases, when the results will not be totally correct.

The result of this work is also only a static approximation, and dynamic methods are of higher importance for efficient systems following the current load and approximating the future loads.

III. Conclusion

This is a work under progress, and the results contained in this paper are still to be refined, and the model has to prove its viability, both requiring an enormous amount of benchmarking. There are still many unknown correlations, and several possibilities to refine the model.

Acknowledgement

I would like to acknowledge my students working on the field of virtualization benchmarking – András Dóczi, László Laposa, Balázs Simon – whose work, measurement data, exploration of virtualization technologies helped in trying to assemble the model discussed in the paper.

References

[1] Gerald J. Popek and Robert P. Goldberg (1974), "Formal requirements for Virtualizable Third Generation Architectures" *Communications of the ACM 17 (7), pp. 412-421*

[2] Ildikó Váncsa, "Construction of a virtualization-based high-availability server" *BUTE VIK TDK conference 2007*

[3] Gunjan Khanna, Kirk Beaty, Gautam Kar, Andrzej Kochut, "Application Performance Management in Virtualized Server Environments," *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, Vancouver, BC, pp. 373-381*

CLASSIFICATION OF IMAGES IN BIOMEDICAL PUBLICATIONS

Márta ALTRICHTER
Advisors: Gábor HORVÁTH, Bill ANDREOPOULOS

I. Introduction

In the past years the number of publications in biomedical literature was rapidly growing. The number of publications for example in the open accessed database of BioMed Central (BMC) shows a quadratic tendency [1]. As a result the importance of good query systems is increasing too.

Images in publications of biomedical literature are frequent and often provide important details to the reader, while this high semantic content is hard to be indexed, described by automatic methods. The existing approaches to image annotation for reader queries are manual annotations, text based search and content based image retrieval (CBIR) [2].

Manual annotation has obvious disadvantages as the number of images to annotate grow: human time consumption, more annotators can have different views on the same image (an example system is Google Image Labeler). Text based search engines work on querying the title of the image (new techniques try to extend this text mining to the paragraph where the image was referenced too [1]). CBIR systems try to index images based on image features like colors, texture, ... Query-by-image systems use the feature vector of the query image and search for images with similar feature vectors. The presently available CBIR systems mainly work on general image databases, while biomedical databases are harder as images are often very similar while representing totally different content. Hybrid systems try to merge information acquired from images and from the text too.

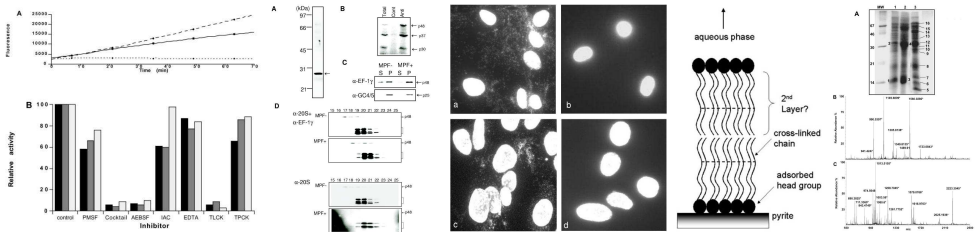
Our main aim is to create a system incorporated to GoPubMed, where the user can search for an image based on text query and by choosing between predetermined image classes. An example query would be: 'fruit fly evolution' search string with the class 'graph'. This query would be expected to return images which are graphs and somehow correlate to 'fruit fly evolution', like a population change graph of a fruit fly community.

In this paper we focused on image classification based on only image features extracted. Moreover we started to propose a way to deal with images containing more panels having different type of images to get closer to the main aim.

II. Image Classes

We downloaded 12598 articles of PubMed publications in years 2000 to 2005. These articles contained 126865 images (of which some are duplicates or thumbnails). Based on our overlook on the images and on the previous work done by Rafkind, Lee, Chang and Yu [3] we decided on the following classes to distinguish between as a first step:

1. Graph: all kind of charts (bar, plot, pie, ...), including hierarchical charts and hand made ones.
2. Gel: images showing chromatographic experiment results.
3. Thing: images, photographs of existing objects like a cell (microscopic), tissues, organs or photo of lab equipment, ... Subclasses: a, Microscopic: as approximately half of the Thing images turned out to be microscopic images it is a feasible way to experiment with the division of Thing to Microscopic and Non-microscopic subclasses. b, Non-Microscopic.
4. Model: any model of a biological process, experiment model, protein sequence or higher protein structure representations, any model of an equipment, drawings ...



(a) Graph (b) Gel (c) Thing (d) Model (e) Gel-Graph

Figure 1: Image Taxonomy

5. Mixed: the black sheep images are the mixed images. These images have more panels on 1 image, and these panels (unlike in Graph type when 2-3 or more graphs on 1 image) differ in types. Most common are Graph-Gel, Graph-Model and Graph-Thing pairing. Images containing 3 or more different types is very rare.

Different from the authors of [3] we plan to have different classes for all of the pairs of Mixed category, as common sense suggests that when a person is looking for a graph of a 'fruit fly' he intends to find those images too which may contain some irrelevant information but having a graph on it. In this case the person wouldn't choose a query separately as Mixed images with the 'fruit fly' text based search as this kind of query could return absolutely irrelevant results for the user like a Gel-Model image too, while the user tries to find Graphs of 'fruit fly'. On Fig 1. you can see some examples for the biomedical image classes.

III. Feature extraction and preliminary image classification

We decided on extracting image features to use machine learning to see preliminary image classification. Based on experimenting with different features and using up the suggestions in [3] we decided on to use intensity histogram features and edge based projection features. At extraction of these features, first the RGB images are converted to grayscale by eliminating the hue and saturation information while retaining the luminance.

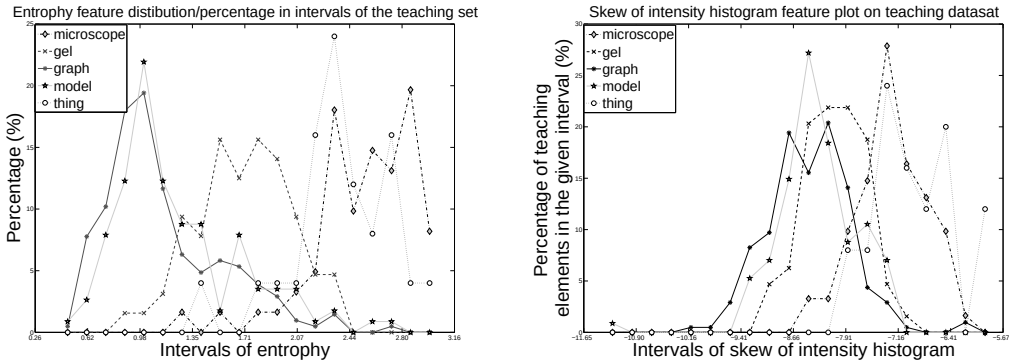
For the extraction of the histogram features the histogram containing the 0 to 256 grayscale value is created and normalized by dividing the sum of all the bins. The mean, variance, entropy, kurtosis and skew of this plot is determined. Fig. 2. shows what percentage of the teaching set is in certain intervals for some of these features.

For the edge based projection features first the grayscale image is filtered by a Sobel operator, than the image is projected onto x-axis (summed vertically) and onto y-axis (summed horizontally). The two plots are again normalized, and the entropy, mean, variance, kurtosis and skew calculated again. Altogether histogram and edge-based features result in 15 features.

To classify the images into the classes mentioned in Section II. we constructed a neural network with 10 neurons in hidden layer and 4 at output layer. Each of the output neurons gives a classification probability of the image being 1 of the 4 classes (Graph, Gel, Model, Thing) based on the feature vector. The desired output vector is a vector having 1 or 0s, e.g a Graph-Gel image has vector $[1, 1, 0, 0]$.

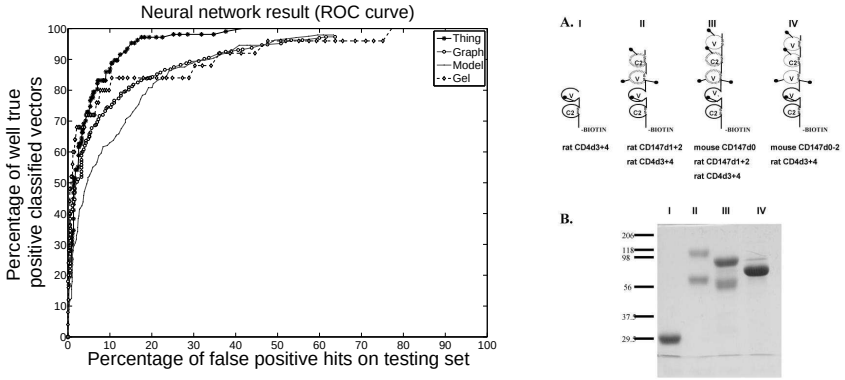
The network was taught by 93*4 teaching vectors, testing set contained 25*4 vectors for early stopping, and the remaining 886 vectors were used for validation on how good the taught network's classification ability is.

We observed a significant increase in the precision by the introduction of the edge based projection features, most probably cause these features quite well characterize the x and y axis line's in Graph images for example. Also we have to note that based on these features Thing images already quite well separate, most probably cause those have very distinct characteristics like dense filled square like



(a) Entropy of intensity histogram (b) Skew of intensity histogram

Figure 2: Some examples of feature distribution on the teaching sets



(a) ROC curve (b) Image with panels

Figure 3: ROC curve of NN results and an example for a difficult paneled Gel-Model image

panels (microscope panels for example), while the worst performance is on models, as model images have big variance from hand drawn images to UML box diagrams, or protein sequences involving lot of characters and connecting lines. The ROC curve represents the TP/FP percentage ratio of the neural network's results on the validation set. To create this ROC curve a threshold from 0 to 1 is applied on each output neuron's result, and if the output number is higher than the threshold the given input sample is classified as one of the class represented by that neuron. Note that in this way one input sample can become member of more classes, which is important in order to handle Mix type images. See Fig. 3(a).

IV. Panel separation

For any chance to handle Mixed Images an algorithm is needed to separate the sub-images, many times called panels on the biomedical image. Unfortunately in general biomedical databases images containing several panels are not always of compact panels (like microscope image panels of Fig. 1(c)) but sometimes a panel does not have a distinct square like boundary as in Fig. 3(b).

Panel separation starts by converting RGB images to grayscale, than finding a background threshold by Otsu's method and converting the greyscale image to binary image. On this binary image the bounding boxes of separated objects are calculated. The shortest distance between each object pairs' boundary boxes is calculated. As not the distance between object centroids are used but the distance measured between boundaries the standard clustering methods (K-means, subtractive clustering, ...) cannot be applied.

(a) Original image (b) It = 0 (c) It = 1 (d) It = 2 (e) It = 4

Figure 4: Iteration process for object clustering to find panels: small objects are moved toward the big grey ones, overlapping objects are brighter

A modified gravitational clustering algorithm of the one suggested in [4] is used as a heuristic way to find out which smaller bounding boxes are together nonetheless (like in most cases the numbers of x and y axis on graphs are separate objects, that need to be clustered in with the big bounding box of the graph to form the actual panel).

Gravitational forces are calculated on each object. There is a normal force on o_1 exerted by an other object o_2 on it: $F_g = G * m_{o_2}/(dist_{o_1-o_2})$, where G is a gravitational constant decaying with each iteration, and m_{o_2} is the mass (object size) of o_2 . There is a repulsive force on o_1 exerted by an other object o_2 if both objects are big: $F_r = -2 * G * m_{o_2}/(dist_{o_1-o_2})$, in this way the big bounding boxes which most probably belong to 2 different panels try to push each other away. And a third repulsive force on o_1 exerted by an other object o_2 is when o_2 is a well filled object, meaning that most of it's bounding box region is not filled with background pixels but useful information. This is to make the filled regions like microscope images push other objects away from them. $F_r2 = -o_{2coverage} * G * m_{o_2}/(dist_{o_1-o_2})$, where $o_{2coverage}$ is a value from 0 to 1 describing how big ratio of the bounding box is non-background pixel. All the forces applied on an objects are summed up making the resultant, than the object is moved in the direction of resultant divided by its own mass m_{o_1} (so heavier objects move slower): $s = F_sum/m_{o_1}$. All movement is calculated for all the objects, than G is decreased and the process is iterated again till there is movement or the maximum iteration limit is reached. One process is illustrated on Fig. 4.

V. Conclusions

We created a system giving preliminary classification of images. We proposed a new way for panel separations on images made up of small sub-figures.

There is still a need to improve the features on which the classification is based, like incorporating color (RGB) information too. An extensive research of the panelization is needed, to determine when the heuristic fails, and the best gravitational constants to give general good results. Future work is to combine image based classification results with the text mining based classification detailed in Strobert's work [1].

References

[1] H. Strobert, "An image retrieving system for biomedical literature using text-mining and ontologies," M.S. thesis, Technological University of Dresden.

[2] Y. Eui and T. S. Huang, "Image retrieval: current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*, 10(1):39–62, Mar. 1999.

[3] B. Rafkind, M. Lee, S.-F. Chang, and H. Yu, "Exploring text and image features to classify images in bioscience literature," in *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology HLT-NAACL 06*, New York City, USA, June 73–81 2006.

[4] J. Gomez, D. Dasgupta, and O. Nasraoui, "A new gravitational clustering algorithm," in *Proceedings of the Third SIAM International Conference on Data Mining 2003*, pp. 83–94, 2003.

EDGE DETECTION ALGORITHM FOR CAPILLARY MICROSCOPIC IMAGES

Gábor HAMAR
Advisors: Gábor HORVÁTH (BUTE-MIT),
Zsuzsanna TARJÁN (Polyclinic of the Hospitaller Brothers, Budapest),
Tibor VIRÁG (Kokabura Ltd.)

I. Introduction

Capillary microscopy is the examination of the smallest vessels of the human organ, the capillaries. The peripheral blood circulation is very sensitive for certain illnesses e.g.: autoimmune diseases, diabetes, etc. In many cases the deformations in the blood circulation can be observed prior to other symptoms, therefore capillary microscopic tests play an important role in the early detection of these diseases [1].

Today the main problem is that there is no cheap, easily accessible instrument, which is capable not only of the image or video recording, but has a support for computer aided evaluation. This is very important because the exact and objective evaluation requires much more time than the microscopic image capture itself, and this is why quantitative measures are rarely used.

The first problem of image evaluation is the detection of capillaries, this is essential for any further image processing steps. We have to achieve high hit rate with low number of false positive hits, despite of the low image quality. In our presentation we will introduce an edge-detection method which can solve this problem with a relatively high performance.

II. Materials and Methods

A. Capillary microscopic images

The capillary microscopic pattern of a healthy patient was examined by many researchers, hence it is precisely defined by the medical literature. As one can see in Fig. 1(a) the vessels are arranged into rows, they are regular hairpin shaped, with the same orientation. A capillary loop has two parallel stems: a thinner called arterial section, and a wider called venous section. They are connected with a winding part called apical section.

The most important parameters that can be extracted from the picture: the arrangement of the vessels, sizes of the hairpin (length, distance of the two stems, diameters), shape of capillaries, linear density, occurrence of micro haemorrhages and visibility of the lager vessels: the SVP (Subpapillary Venous Plexus).

In certain diseases the healthy pattern changes. In many cases the regular arrangement breaks up. If the vessels become dilated, they are called giant- or mega-capillaries according to their size. The hairpin shape can also be changed: the medical literature classifies the modified shapes into the following groups: meandering, bushy, ball, tortuous and ramified. The linear density decreases in general, in certain cases micro haemorrhages can be observed, and the visibility of the SVP increases.

B. The method of image recording

There is a generally accepted method for capillary microscopic examinations, hence we have also followed this method [2]. Our present database was created with a light microscope. We used paraffin oil in order to increase the transparency of the skin, and a light source (intralux 6000) with cold light. The direction of the light was approximately 45°.

C. *Edge detection*

As it can be seen in Fig. 1(a) the capillaries do not have sharp edges, the image is very blurred and noisy. These properties are mainly due to the recording method, because capillaries are observed through the skin. Classical edge detectors such as Sobel, Laplace and Canny operators have given bad result. It was very hard to separate real edges from noise, and edge detectors found not only the borders but the whole area of the capillaries because the intensity changes continuously in perpendicular direction to the capillary. Considering these properties we decided to search not the border but the centre line. It can be located more precisely and more robustly, with the algorithms described later.

The whole process of edge detection can be divided into tree parts: amplification, classification and edge connecting. In the first detection step, filters are used which have a high response in the points of the edge. For the construction of such filters first we need a definition for the edges. Usually intensity discontinuities or discontinuities in the first derivatives are treated as edges, therefore the filter is a high pass filter for example a gradient type operators.

In the classification step we need to categorize the pixels into two categories: edge or non-edge pixels. This classification is done according to the response of the amplification operator and the position of the pixel. This step always requires one or more threshold levels. Finding the optimal threshold level is a hard problem, thus there are many different approaches. None of them is optimal in all of the cases, hence we always need to take into account the properties of the filtered image. It is possible, that no acceptable threshold level exist. On low signal to noise ratio images it can happen that the response of the filter to noise becomes greater than the response to real edges.

The result of the classification step is a set of individual pixels which may or may not belong to the same edge. In this step we have to connect these individual points into one or more connected lines. The result is on or more sequence of pixel positions, where each sequence represents an edge [3].

We developed an image processing system to solve these three steps of edge detection. The three steps of the proposed algorithm will be described in the following subsections.

D. *Edge amplification*

As we saw capillary microscopic images are noisy and blurred, therefore the signal-to-noise ratio of the images are low. The problem with classical edge detectors is, that they make the decisions based on local information. One solution to this problem is to increase the number of pixels that take part in the decision process. Classical signal processing theory achieves this by increasing the size of the edge detection operator. This method has a limitation because as we increase the size of the operator the resolution decreases. Further improvement can be achieved in the performance with using a set of directional operators, which compute their response based on the pixels of a long and narrow rectangle. The first problem of this idea is that we have to increase the number of operators, which increases the computational burden. Secondly edges on a real image may not be straight on a very long distance.

One solution to this problem is to compute the response of the edge operator by using the pixels near the edge contour, instead of the pixels of a rectangular area. This is not a straightforward solution because the contour of the edge is not known before the edge detection, so we have to estimate it, or perform a search in the space of possible contours during the edge detection. Many techniques have been developed based on this idea, such as sequential edge detection algorithms. Our algorithm is based on the same idea but uses a different approach in the realisation.

The method, first extracts the local information from the image, and then it makes decisions based on global measures. We use gradient type operators, only for estimating the direction of the edges in every pixel of the image. We compute the two neighbour pixels for every pixel along an edge by using the calculated direction. This two neighbouring pixel can be calculated for every pixel. In case of non-edge pixels, the result will be faulty, but then we consider it as a random value. We have an assumption: if a pixel is located on one of the edges of the image then the two neighbours will be on the same edge with high probability. Now we can regard the image as a directional graph, where

every pixel is a point in the graph, and the graph edges point from a certain pixel to its two neighbours. With the help of this graph we can make the definition of edges using the concepts of the graph theory: edges are strongly connected subsets of the graph i.e. sets of points that have many edges with both of the starting and the end point in the set. For finding the strongly connected components there are many procedures. The proposed algorithm is based on a Markov-chain model [4]. This procedure calculates a weight for every pixel which has a high value if the pixel belongs to a strongly connected component. This weighs are displayed as an image in figure 1(b).

E. Pixel classification

The classification step categorizes the image pixels into two subsets: edge pixels and non-edge pixels. This is done by thresholding the output of the filter. A hysteresis threshold is used to avoid streaking. This phenomenon results from the fact that real edges are non-homogeneous. It is possible that the operator output is sometimes above, sometimes below the selected threshold level, due to the noise. Hysteresis threshold uses two threshold levels. If a pixel intensity is under the lower level it is classified as a non edge pixel, if it is over the higher level it is classified as an edge pixel. The remaining pixels with the intensities between the two levels are treated as edge pixels only if they are connected to one ore more pixels that are already classified as an edge pixel.

Our edge amplification procedure has an advantage: the resulted pixel values do not directly depend on the original intensities, but on the relation of pixels. This property results in a more uniform output, which enables us to use global threshold levels for whole image. The threshold levels can be selected based on the histogram of the filtered image.

F. Connecting edges

In the last step of the detection our task is to connect individual pixels to one or more connected lines. A line is represented by a sequence of pixel coordinates. As it can be seen in Fig. 1(b) after the amplification the resulted lines are not continuous lines. There is a large variation in the intensity of edge pixels. We tried many approaches: smoothing, ridge detection, morphological operators, skeletonisation. The problem with these solutions is, that the variation is too high along the edge contour. There is no continuous ridge in the line even after a strong smoothing. Morphological operators and skeletonisation are very sensitive to noise, hence the result is not a smooth line and it is highly dependent on the selected threshold level.

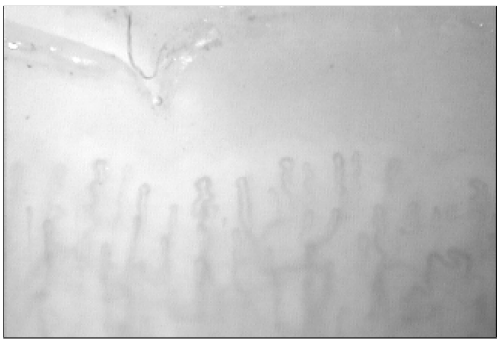
For the human observer the result looks like a continuous line, despite of the high variation of the intensity. We realized that these intensity variations cannot be compensated with rectangular smoothing operators, but line fitting is very effective and stable. We used a least squares method, for fitting lines to the pixels of a small rectangle of the image.

Once the directions of the lines are known the whole line can be detected, starting from one pixel by walking along the edge. The procedure makes steps from an initial position. First it calculates the direction of the edge in the actual position then finds the next edge point in that direction. The resulted point is the starting point of the next step. This procedure is continued until we are moving on edge pixels, according to the result of the classification.

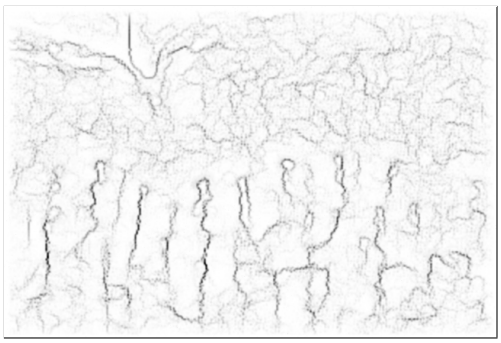
The starting points are also selected based on the classification method. First we choose the edge pixel with the highest intensity value. We start the walking from this point and when the whole line is detected we delete all of the pixels which are near to the detected contour, by setting their intensity value to 0. We continue the procedure by selecting the next starting point. This can not be the same pixel as the previous starting point, because that has been already deleted.

III. Experimental results

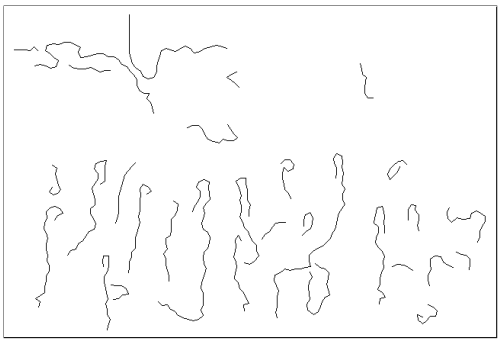
Fig. 1 shows the output of the procedure on a typical capillary microscopic image: 1(a) is the original image, 1(b) is the output of the amplification algorithm and 1(c) shows the connected edges.



(a) Original image



(b) Edge amplification



(c) Detected lines

Figure 1: Result of the edge detection

It is difficult to precisely measure the efficiency of the algorithm, because there is no exact reference with which it could be compared, we can only use the opinion of a human observer. We have applied this algorithm on 50 images, with variable quality. The results are compared with the human's opinion. The vessels detected by the algorithm were 91% of the vessels which are seen in the image.

IV. Conclusion

We have introduced a novel method of edge detection, which uses an edge definition different from traditional edge detectors. This definition is not based directly on local properties of the image, but utilises the relation between pixel positions, which can be calculated using local properties. We have developed and tested our solution on a special image set, but our method can be easily modified to detect other types of edges.

Acknowledgement

This research was supported by the EU, GVOP grant 3.3.3-05/2.-2006-01-0130/3.0.

References

[1] A. Bollinger and B. Fagrell, Eds., *Clinical Capillaroscopy*, Hogrefe & Huber Publishers, 1990.

[2] Zs. Tarján, É. Koó, P. Tóth, and I. Ujfalussy, "Capillary microscopic examinations (kapillármikroszkópos vizsgálatok)," *Magyar Reumatológia*, 42:207 – 211, 2001, (In Hungarian).

[3] P. H. Eichel, "Method of detecting intensity edge paths," 1990, United States patent, No: 4,910,786.

[4] G. Hamar, G. Horváth, Zs. Tarján, and T. Virág, "Markov chain based edge detection algorithm for evaluating capillary microscopic images," in *IFMBE Proceedings: 11th Mediterranean Conference on Medical and Biological Engineering and Computing*, vol. 16, Ljubljana, Slovenia, June 2007.

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. It is available in many countries including Hungary. Unfortunately such screening programs generate a vast amount of pictures to be analysed 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. Lung contour detection

Several methods have been developed to search the contour of the lungs. The problem is not easy 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 [1]. They start from an average contour and move contourpoints to fit the actual shape in each iteration. The measure of the goodness is an energy function of two parts. One is responsible for the global shape (for example by avoiding high curvature contours) and the other fits the shape locally, usually by finding the maximum of some kind of gradient.

These learning methods need great number of pictures evaluated by experts. Drawing the contours in hundreds of images is a big work and thus usually only a limited number of denoted pictures are available for the algorithms. We used the publicly available reference image database of 247 radiographs by the Japanese Society of Thoratic Radiology [2]. The delineation of these pictures was made by the research group of van Ginneken [3].

ASM [4] can be seen as an extension of snakes. It generates the global and local part of the energy function automatically from a statistic model and defines a search algorithm too.

It groups contour landmark points of each training picture into one vector and applies PCA on them. The first few eigenvectors belonging to the largest eigenvalues are chosen to describe the shapes. The actual contour is computed from the chosen eigenvectors and the mean shape. The weights of the eigenvectors are limited and this gives a restriction on the allowed shapes. Procrustes

analysis [5] can be used before the PCA to align shapes. This makes the shapes independent of position, orientation and size. An example can be seen on Figure 1.

The local fitting is done by building gray-level appearance models at every contour landmark point. These are built from normalized gradient vectors sampled from both sides of the line perpendicular to the contour. Mahalanobis distance [4] is used to choose between different profile candidates.

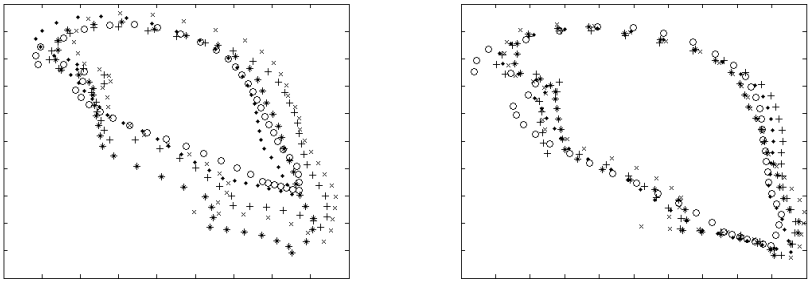


Figure 1: Lung contours without alignment (on the left) and after Procrustes alignment (on the right)

To make the search space smaller, landmark points are moved perpendicular to the contour one by one and the best fitting to the local appearance model is chosen. After a few iterations this process usually converges to a contour.

All these steps are repeated at different resolutions, starting from the coarsest. At a finer resolution the search is continued from the previous contour result. This multiresolution method helps keeping both the computational cost and the contour error low.

III. Results and Conclusion

Test runs show that ASM gives good results compared to ad-hoc algorithms. Over 80% of the test pictures were evaluated without major errors. In these cases the contours were not only a rough approximation of the ideal solution, but they were highly accurate. Most of the errors occur at the early phase of the algorithm, at coarser resolutions.

The solution for this problem could be to use information of the fine resolution pictures too at the early phases or to use more complex texture information instead of the simple gradients.

Another way to improve results is using active appearance models (AAM). AAM combines texture information of the object with the shape information of the contour. This combination highly increases the robustness of the algorithm at the price of higher computational costs.

References

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

[2] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, K. Doi, "*Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules,"* American Journal of Roentgenology 175, 71-74, 2000.

[3] B. van Ginneken, M.B. Stegman, M. Loog, *"Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database,"* vol. 10, pp. 19-40., 2006.

[4] 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.

[5] 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.

ROI SELECTION IN MICROCALCIFICATION DETECTION

László LASZTOVICZA
Advisor: Béla PATAKI

I. Introduction

Microcalcifications are small bright spots and they are usually grouped into clusters in mammographic images. An image can contain more than one cluster but still the area of the clusters is far smaller than the area of the breast. The aim of ROI (region of interest) selection is to select suspicious areas which can contain microcalcifications for further analysis. A proper ROI selection method can increase the overall processing speed and more importantly it can decrease the number of false positive detections. Microcalcification detection algorithms use different strategies to select ROIs. In [1] a wavelet and fuzzy c-means based method is presented. A wavelet based method using an adaptive homomorphic enhancement filter can be found in [2], other classical filter based method in [3] or morphological filtering in [4]. Another strategy can be subtracting the background and using a threshold to detect suspicious pixels and regions [5]. A relevance vector machine based approach is used in [6] and a method applying ant colony optimization and genetic algorithms is presented in [7]. Texture based methods are also exist [8, 9] with rather good performance.

In the present paper we discuss a method for ROI selection and give some modifications in order to improve its performance.

II. Surrounding region dependence method

A. Surrounding region dependence matrix

Surrounding region dependence method (SRDM) is a texture based method for detecting ROIs [8, 9]. For every image pixel (x, y) in the image plane $L_x \times L_y$ we define three windows which give us two regions R_1 and R_2 (see Figure 1). Then we define a matrix using a threshold q,

$$M(q) = [\alpha_{ij}], \quad 0 \le i \le m, \quad 0 \le j \le n, \tag{1}$$

where m and n are the number of pixels in regions R_1 and R_2 and

$$\alpha_{ij} = \#\{(x, y) \mid c_{R_1}(x, y) = i \wedge c_{R_2}(x, y) = j; (x, y) \in L_x \times L_y\}. \tag{2}$$

The counters c_{R_1} and c_{R_2} define the number of those pixels in the surrounding regions R_1 and R_2 where the difference in the intensity is greater compared to the central pixel and defined as,

$$c_{R_1}(x, y) = \#\{(k,l) \mid (k,l) \in R_1 \wedge [S(x, y) - S(k,l)] > q\} \tag{3}$$

$$c_{R_2}(x, y) = \#\{(k,l) \mid (k,l) \in R_2 \wedge [S(x, y) - S(k,l)] > q\} \tag{4}$$

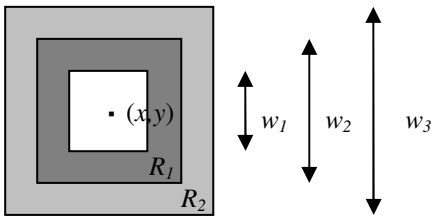


Figure 1: Surrounding region

The α_{ij} elements of the matrix $M(q)$ are the number of those pixels in the image whose difference in the intensity value compared to the pixels in the surrounding regions is greater than the threshold q.

B. Feature selection

Let N is the sum of the elements of the matrix $M(q)$, and $r(i,j)$ is the reciprocal of the elements,

$$N=\sum_{i=0}^{m}\sum_{j=0}^{n}\alpha(i,j);\ r(i,j)=\begin{cases}\dfrac{1}{\alpha(i,j)}, & \text{if } \alpha(i,j)>0,\\ 0, & \text{otherwise.}\end{cases}$$

Then following [8, 9] the features shown in Table 1 can be extracted from the matrix $M(q)$.

Table 1: Features to be extracted from $M(q)$

Horizontal Weighted Sum	Vertical Weighted Sum
$HWS=\frac{1}{N}\sum_{i=0}^{m}\sum_{j=0}^{n}i^2r(i,j)$	$VWS=\frac{1}{N}\sum_{i=0}^{m}\sum_{j=0}^{n}j^2r(i,j)$
Diagonal Weighted Sum	Grid Weighted Sum
$DWS=\frac{1}{N}\sum_{k=0}^{m+n}k^2\left(\sum_{\substack{i=0\\ i+j=k}}^{m}\sum_{j=0}^{n}r(i,j)\right)$	$GWS=\frac{1}{N}\sum_{i=0}^{m}\sum_{j=0}^{n}ijr(i,j)$

For each ROI a matrix is computed then the four features are extracted. The feature vectors serve as input to a neural network which performs the classification of the ROI either as negative or positive.

III. Improving the SRMD method

Though the method itself provides fairly good results I introduced some modifications to further improve the performance.

A. Image enhancement and feature extraction

First I introduced an image enhancement step. The enhanced image was then used as input to the SRDM method instead of the original image. The enhanced image can also be considered as a feature image and can be used in the detection of microcalcification clusters provided the ROI was evaluated as positive. The image enhancement is as follows. The input ROI is first filtered with an averaging filter then the filtered ROI is subtracted from the original one. For every pixel the following neighborhood operation is carried out

$$S_{diff}(x,y)=S(x,y)-\frac{1}{n}\sum_{i,j\in W}S(i,j), \tag{5}$$

where W is a fixed sized window or kernel with size n centered around (x,y) with intensity $S(x,y)$.

From some pilot measurement the kernel size was determined in 21 pixels, however for some settings other kernels also had a similar performance. Therefore I introduced a simple modification. As the brightest pixels are around the center of the microcalcifications at least locally, it is likely that the average will decrease as the kernel size is increasing. For every pixel a series of average values can be determined using different kernel size (e.g. 13, 15, 17 … pixels) according to the size of the individual microcalcifications. Then for every pixel I chose the feature value given by Eq. (5) where the average reached minimum.

B. Introducing new features

In order to further enhance the performance of ROI selection I introduced three more features that are not related to the SRDM matrix. The features are extracted from the ROI after the enhancement

step. For each ROI two sums were computed. First summing the intensity values by rows and second summing them by columns. That gives us two vectors (e.g. vertical and horizontal sums): $F_1(r) = \sum_c S(r,c)$ and $F_2(c) = \sum_r S(r,c)$. Then two features are simply the standard deviation of the vertical and horizontal sums F_1 and F_2. The third feature was simply the standard deviation of the intensity values in the input image. Analyzing the new features they show good characteristics in discriminating capability.

IV. Experiments and results

For the experiments 200 positive and 200 negative ROIs were extracted from images of the DDSM database [10]. In the experiments I used a 10-fold cross-validation setting in order to determine the optimal setting of the parameters for the method and to measure the performance of the algorithm and the proposed improvements. Neural networks were used as classifiers. The neural networks had one hidden layer with 8 neurons and with tangent hyperbolic nonlinearity. The number of inputs was determined by the number of features. For the SRMD method the number of input features is 4 while for the extended method it is 7.

The parameters of the SRDM method are the size of the windows (w_1, w_2, w_3) and the threshold q. For the proposed improvements the effect of the new features (*input – std* stands for the new features) and the size of the kernel (w_4 – *avg* stands for the procedure for choosing feature value presented in section III) for the feature extraction/image enhancement step. I also examined the case when the samples in the training set containing microcalcifications are doubled to increase the error if a sample containing microcalcification is misclassified (*tr_set*) and the case when the resolution of the image is decreased (*res* - size of an input ROI in pixels). The classification accuracy for the cross-validation is also presented (*acc*). Table 2 shows the 4 best results and the best result for the original method (in light gray). The meaning of the columns is explained above. In Table 2 the results are given as the area under the ROC curve (A_z) and the 95% confidence interval for the A_z value is also given

Table 2: The best results for different parameter settings

w_1	w_2	w_3	q	w_4	res	tr_set	input	A_z	confidence (95%)
3	7	11	11	21	256x256	-	SRDM + std	0.959	0.921 to 0.982
5	7	9	8	avg	256x256	-	SRDM	0.958	0.920 to 0.981
3	7	11	8	avg	256x256	-	SRDM	0.957	0.918 to 0.980
3	5	7	7	21	128x128	-	SRDM + std	0.954	0.915 to 0.979
3	7	11	11	-	256x256	-	SRDM	0.949	0.909 to 0.975

. The ROC curves for the best result reached by the original and the modified method are shown in Figure 2.

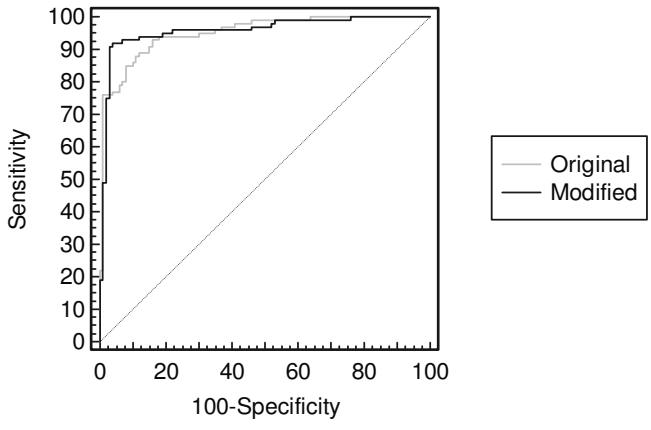


Figure 2: ROC curves for the best results

As it can be seen from Table 2 the results are very similar and there is no statistically significant difference in the results. Still they show that by using the introduced new features and using enhanced images as inputs the performance can be increased. It is also interesting that fairly good result can be reached even when the resolution of the input ROI is halved which can make the processing faster. The best results were reached by the improved method which proves its viability.

In Figure 2 though there are no significant differences between the ROC curves it can be seen that the modified method reached higher sensitivity at higher specificity therefore it improved the original methods by decreasing false positive detections.

V. Conclusion and discussion

The proposed modifications to the original SRDM method were using enhanced or feature images as input images and introducing new features which expresses the variability of pixel values in rows, columns and in the image itself. As it can be seen from the results the modified method is capable of reaching better performance than the original one. Future experiments are required to prove that on larger datasets. It is also true however, that there is no statistically significant difference between the best results at this point, therefore either the number of samples is too low to reach any significant conclusion or there is no real difference between the methods and the selection of parameters. The latter suggests using combined classifiers as the difference between the different parameter settings is not significant – they are giving good results in themselves – but they are mistaken in possibly different cases. Another way can be to find adaptive selection algorithms for the different parameters. Finally the two ways can be combined into a hierarchical multiple classifier system.

References

[1] S. Sentelle, C. Sentelle, and M. A. Sutton, "Multiresolution-based segmentation of calcifications for the early detection of breast cancer," *Real-Time Imaging*, vol. 8, no. 3, pp. 237–252, 2002.

[2] H.-K. Kang, S.-M. Kim, N. N. Thanh, Y. M. Ro, and W.-H. Kim, "Adaptive microcalcification detection in computer aided diagnosis," in *Computational Science - ICCS 2004, 4th International Conference, Kraków, Poland, June 6-9, 2004, Proceedings, Part IV*, ser. Lecture Notes in Computer Science, M. Bubak, G. Albada, P. Sloot, and J. Dongarra, Eds., vol. 3039. Springer-Verlag Berlin Heidelberg, 2004, pp. 1110–1117.

[3] P. Pettazzoni, G. Pallotti, M. Mattina, and C. Leoni, "Computerized detection of clustered microcalcifications: a modular approach with non-linear filters," *Medical Hypotheses*, vol. 56, no. 4, pp. 442–447, 2001.

[4] J. C. Yang, J. W. Shin, G. S. Yang, and D. S. Park, "Microcalcifications detection in digital mammogram using morphological bandpass filters," in *Computational and Information Science, First International Symposium, CIS 2004, Shanghai, China, December 16-18, 2004. Proceedings*, ser. Lecture Notes in Computer Science, J. Zhang, J.-H. He, and Y. Fu, Eds., vol. 3314. Springer-Verlag Berlin Heidelberg, 2004, pp. 492–497.

[5] A. Papadopoulos, D. I. Fotiadis, and A. Likas, "An automatic microcalcification detection system based on a hybrid neural network classifier," *Artificial Intelligence in Medicine*, vol. 25, no. 2, pp. 149–167, 2002.

[6] W. Liyang, Y. Yongyi, R. M. Nishikawa, M. N. Wernick, and A. Edwards, "Relevance vector machine for automatic detection of clustered microcalcifications," *Medical Imaging, IEEE Transactions on*, vol. 24, no. 10, pp. 1278–1285, 2005.

[7] K.Thangavel, M.Karnan, R.Sivakumar, and A. K. Mohideen, "Ant colony system for segmentation and classification of microcalcification in mammograms," *The International Journal of Artificial Intelligence and Machine Learning*, vol. 5, 2005.

[8] J. K. Kim and H. W. Park, "Statistical textural features for detection of microcalcifications in digitized mammograms," *Medical Imaging, IEEE Transactions on*, vol. 18, no. 3, pp. 231–238, 1999.

[9] J. K. Kim, J. M. Park, K. S. Song, and H. W. Park, "Detection of clustered microcalcifications on mammograms using surrounding region dependence method and artificial neural network," *The Journal of VLSI Signal Processing*, vol. 18, no. 3, pp. 251–262, Apr. 1998.

[10] M. Heath, K. Bowyer, D. Kopans, R. Moore, and W. P. Kegelmeyer, "The digital database for screening mammography," in *Proceedings of the Fifth International Workshop on Digital Mammography*, M. Yaffe, Ed. Medical Physics Publishing, 2001, pp. 212–218.

VERIFICATION OF MODEL TRANSFORMATION

Ákos HORVÁTH
Advisor: Dániel VARRÓ

I. Introduction

As model driven software development (MDSD) pervaded the safety critical (SC) and dependable system development processes, higher demands rose on their design, maintenance and certification. Based on high-level modeling standards (e.g. UML), MDSD separates application logic from underlying platform technology by using platform independent models (PIM) to capture the core functionality of the target system, and platform specific models (PSM) to specify the target system on the platforms (SCJava, ADA, C/C++). PSMs and platform-specific source code are automatically generated from PIM and PSMs, respectively, by using model transformation (MT) techniques.

A critical problem is related to the correctness of model transformations is to guarantee certain semantic properties to hold after transformation execution. For instance, when transforming UML models into Petri nets, the results of a formal analysis can be invalidated by erroneous model transformations as the systems engineers cannot distinguish whether an error is in the design or in the transformation. While there are already a large number of model transformation descriptions [1], we focus on graph transformation [2] (GT) as it is (i) a frequently used mean to capture model transformation and (ii) has thoroughly studied mathematically precise syntax and semantics.

In this paper we introduce our vision for verifying property preservation of graph transformation systems. First, we split the verification process into two levels *design* and *implementation* to separate the precise model from its underlying implementation. The idea is to keep *shape analysis* [3] based abstract interpretation in the design level to verify behavioral attributes for the whole transformation, while on the other hand use Hoare [4] styled code analyzers to verify matching properties of each GT rule used in the transformation. This separation allows to reuse the verification results from the design level in the implementation level in case we can assure the correctness of each used GT rule in the transformation.

The rest of the paper is structured as follows, in Sec. II. we briefly introduce the concept of graph transformation and metamodeling, while Sec. III. proposes our verification approach and finally, Sec. IV. concludes the paper followed by future work.

II. Background

In order to introduce our vision this section briefly introduces the basics of metamodeling and graph transformation.

A. Models and Metamodels

Metamodeling is a fundamental part of model transformation design as it allows the structural definition (i.e., abstract syntax) of modeling languages. More precisely, for the definition of a modelling language the followings shall be given

- the abstract syntax defining the concepts of the given domain and their relations,
- the concrete syntax defining the textual or graphical notations of the concepts,
- well-formness rules defining further constraints for the concepts,
- the formal semantics defining the dynamic behaviour of the models.

In our approach, we use a unified directed graph representation serves as the underlying model of the VIATRA2 [5] framework. This way, graph nodes called entities in VIATRA2 uniformly represent

MOF packages, classes, or objects on different metalevels, while graph edges with identities called relations in VIATRA2 denote MOF association ends, attributes, link ends, and slots in a uniform way. As a summary, nodes represent basic concepts of a (modeling) domain, while edges represent the relationships between model elements.

B. Graph Transformation

Graph transformation (GT) is a rule and pattern-based paradigm frequently used for describing model transformation. A graph transformation rule contains a left-hand side graph LHS, a righthand side graph RHS, and (one or more) negative application condition graphs NAC connected to LHS. A negative application condition is a graph morphism, which maps the LHS pattern to a NAC pattern. In other terms, the LHS and NAC graphs together denote the precondition while the RHS denotes the postcondition of a rule.

The application of a rule to a host (instance) model M replaces a matching of the LHS – which is not invalidated by a matching of the NAC, which prohibits the presence of certain nodes and edges – in M by an image of the RHS.

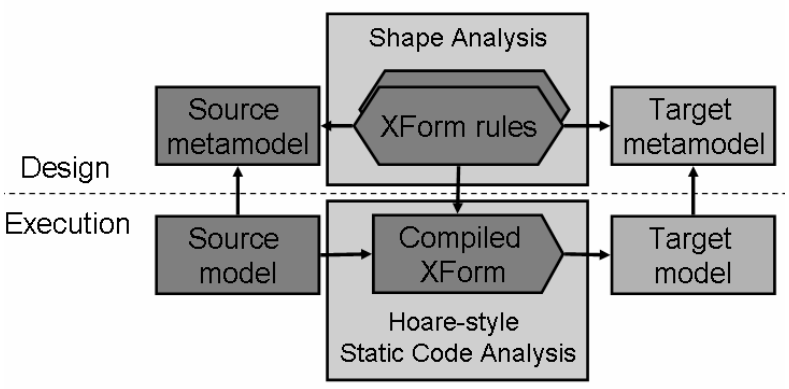


Figure 1: Overview of the approach

Fig 1 gives an overview of our common model transformation process. The model transformation (*XForm rules*) is specified by a number of graph transformation rules. The GT rules are specified with respect to the metamodels of the source and the target metamodel. From these rule specifications, a compiled transformation (*compiled XForm*) is generated. The automatically derived compiled XForm transforms a source model into a target model.

III. Overview of the Approach

Our goal is to verify property preservation for the compiled transformation, meaning that, if a certain property holds in the source model, then after executing the transformation it will also hold in the target model. To do so we separate the verification process into two steps.

- First, we apply shape analysis on the XForm rules to summarize the behavior of a statement on an infinite set of possible rundown states of the GT rules. Shape analysis concerns the problem of determining *shape invariants* for programs that perform destructive updating on dynamically allocated storage. This way correctness of transformation rules applied to *any* model of the specified type can be verified (the concrete instances of the metamodels are irrelevant for the proof).
- Then, as the result of the shape analysis is based on the assumption that the GT rule specification are ”executed” semantically correct, in the second step we focus on the correctness check of the compiled GT rules. As the correctness of the generated compiled code depends on the correctness of the generator itself, which is usually a complex software components that can not

be verified. We use an alternative assurance approach, in which the generator is extended with formal program specification to enable Hoare-style safety analysis for each individually generated GT rule. The crucial step in this approach is to extend the generator to produce all required annotations without compromising the assurance provided by the subsequent verification phase.

A. Design analysis

Shape analysis: In our approach we plan to use the TVLA [6] (Three-Valued-Logic Analyzer), a system for automatically generating a static (shape) analysis implementation from the operational semantics of XForm rules. The small-step structural operational semantics is written in a meta-language based on first-order predicate logic with transitive closure. The main idea is that program states are represented as logical structures and the program transition system is defined using first order logical formulas. TVLA automatically generates the abstract semantics, and, for each program point, produces an abstract representation of the program states at that point. TVLA relies on a fundamental abstraction operation for converting a potentially unbounded structure into a bounded 3-valued structure (logic). 3-valued logic extends boolean logic by introducing a third value 1/2 denoting values that may be 0 or 1. A 3-valued logical structure can be used as an abstraction of a larger 2-valued logical structure. This is achieved by allowing an abstract state (i.e., a 3-valued logical structure) to include summary nodes, i.e., individuals that correspond to one or more individuals in a concrete state represented by that abstract state.

Our initial examples with the TVLA system shows that the mapping of the metamodel to the TVLA is the key for efficient shape analysis generation. On the other hand a surprising experience was that in some case a more precise analysis created smaller shape invariant and thus run faster. Based on [7] this is quite regular in case of 3-valued logic and will have to be extensively studied to create effective mapping of metamodels and GT rules to the analyzing domain.

B. Implementation analyze

Hoare-style platform specific code analyzers: Hoare logic is a formal system to provide a set of logical rules in order to reason about the correctness of computer programs with the rigour of mathematical logic. The central feature of Hoare logic is the *Hoare triple*. A triple describes how the execution of a piece of code changes the state of the computation. A Hoare triple is of the form $\{P\}\ C\ \{Q\}$ where P, Q and C are *precondition*, *postcondition* and *command*, respectively. Based on the concept of pre-/postcondition introduced in the Hoare triple, *design by contract* [8] (DBC or programming by contract) prescribes that software designers should define precise verifiable interface specifications (pre/postconditions) for software components based upon the theory of abstract data types and the concept of a business contract. This means that *contracts* provides semantics to formally describe the behavior of a program module, removing potential ambiguity with regard to the module implementation.

Tools built upon the DBC methodology include the logic of predicate calculus and Dijkstra's weakest precondition calculations. We focused our studies on two of the most widely used frameworks:

- The Spec# [9] programming system is a new attempt having developed at Microsoft Research to extend C# with formally verifiably method contracts in the form of pre-/postconditions as well as object invariants. It consist of (i) the Spec# compiler, which statically enforces non-null types, emits run-time checks for method contracts and invariants, and (ii) the Boogie static program verifier, which generates logical verification conditions from a Spec# program and then uses a built in automatic theorem prover that analyzes the verification conditions to prove the correctness of the program.
- The *KeY* [10] system is a formal software development tool that aims to integrate formal specification, and formal verification of Java programs. The main component of the KeY system is the KeY prover, a semi-automated prover over the Java Dynamic Logic (JavaDL) calculus (with support to Java Modeling Language (JML)). The JavaDL calculus covers the complete

Java Card language, and additionally supports some Java SE features such as multi-dimensional arrays and dynamic object creation. Verification with KeY proceeds by symbolic execution of the Java program being analyzed, where the proof corresponds to a stage during the execution of the program.

Both approaches look promising but does not provide support for: (i) dynamic casting of complex data structures (e.g., arrays), (ii) effective handling of nested loop invariants, (iii) contracts for library functions and finally (iv) user-friendly feedback from proof obligations.

Note that, it is also important to extend the compiled XForm generator in such way that it produces all required annotations (i.e., pre-/postconditions and loop invariants) without compromising the assurance provided by the subsequent verification phase. This is achieved by embedding annotation templates into the code templates, which are then instantiated in parallel by the generator. This is feasible because the structure of the generated code and the possible safety properties are known (from the XForm rules) when the generation is applied. It does not compromise the provided assurance because the annotations only serve as auxiliary lemmas and errors in the annotation templates ultimately lead to unprovable safety obligations.

IV. Conclusion and Future Work

We have presented an ongoing work how graph transformations can be verified with a combination of shape analysis (with TVLA) and static code analyzer (e.g., Spec#, KeY). In the current state of our research, we have studied the boundaries of Hoare-style static code analyzers with respect to complex object navigation (as being the core of transformation implementation). It resulted in state space explosion in case of common implementations of GT rules and have to be further studied to achieve analyzable implementation.

As for the future, we plan to finish formalizing GT rules in 3-valued logic to achieve feasible shape analyze results. Current research in this direction shows that in case of strict metamodels the shape analysis resulted suitable shape invariants for the automatic property checks.

References

[1] K. Czarnecki and S. Helsen, "Classification of model transformation approaches," *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, 2003.

[2] G. Rozenberg, Ed., *Handbook of Graph Grammars and Computing by Graph Transformation, volume 1: Foundations*, World Scientific, 1997.

[3] M. Sagiv, T. Reps, and R. Wilhelm, "Parametric shape analysis via 3–valued logic," in *Symposium on Principles of Programming Languages*, pp. 105–118. ACM Press, 1999.

[4] C. A. R. Hoare, "An axiomatic basis for computer programming," *Commun. ACM*, 12(10):576–580, 1969.

[5] A. Balogh and D. Varró, "Advanced model transformation language constructs in the VIATRA2 framework," in *Proc. of the 21st ACM Symposium on Applied Computing*, pp. 1280–1287, Dijon, France, April 2006. ACM Press.

[6] T. Lev-Ami, R. Manevich, and S. Sagiv, "Tvla: A system for generating abstract interpreters," in *IFIP Congress Topical Sessions*, R. Jacquart, Ed., pp. 367–376. Kluwer, Aug. 2004.

[7] R. Wilhelm, S. Sagiv, and T. W. Reps, "Shape analysis," in *Computational Complexity*, pp. 1–17, 2000.

[8] B. Meyer, "Applying "design by contract"," 25(10):40–51, Oct. 1992.

[9] "Spec#, The Spec# programming system," http://research.microsoft.com/specsharp/.

[10] "The KeY Project, Integrated Deductive Software Design," http://www.key-project.org/.

DESIGN-TIME SIMULATION OF DOMAIN-SPECIFIC MODELS BY INTERACTIVE MODEL TRANSFORMATIONS

István RÁTH
Advisor: Dániel VARRÓ

I. Introduction

In industrial applications (such as embedded system design), domain-specific modeling languages (DSML) are frequently used to model dynamic systems. Futhermore, models are not only used to generate the target application's source code, but also for performing design-time simulation, analysis, validation, and verification using special tools.

Thus, an important issue that needs to be addressed when developing new DSMLs is the precise specification of the semantics of the language. Since this is mostly lacking in current language engineering tools [1, 2], these features have to be implemented using manual coding, which is time consuming and expensive.

In the current paper, we present the new facilities of the ViatraDSM domain-specific modeling framework [3], which aims at bridging the gap by an integrated support for (i) model simulation and (ii) model-to-model transformations. ViatraDSM relies on the underlying VIATRA2 model transformation framework [4] to provide a unified high-level formalism for the specification of dynamic behavioral semantics of a language and to capture model-to-model transformations.

II. Model execution in domain-specific languages

Discrete model execution is applicable to domains where the state-space can be evaluated in discrete time intervals, i.e. changes are *atomic*. While this may also include the discrete approximation of complex continuous time systems such as signal networks, in practise, our approach is targeted at those domains where all concepts can be captured using a bounded (small) number of dynamic entites with each having a well-defined life cycle. This includes "token game" simulations and also systems involving dynamically created object instances (birth-death processes).

Model execution can be either fully automatic or user-guided. User assistance is a requirement in many cases, e.g. when design-time "debugging" is performed by designers. Another source of *interactivity* may be non-determinism, which is frequently present in practical model simulation scenarios. In that case, the user is given a set of choices at a *choice point*, and the execution continues depending on the actual choice of the user.

Therefore, we have chosen to build domain-specific interactive model simulators by using (i) a statemachine formalism to drive user interaction on a very high level of abstraction (Sec. IV.), and (ii) the model transformation language of VIATRA2 to capture atomic simulation steps (Sec. III.), which preserve domain-specific validity constraints.

III. Domain model execution in VIATRA2

First we present how to precisely capture elementary model simulation steps by combining two formal methods to design model simulation for Petri nets. For this purpose, a brief overview is provided to the transformation language of VIATRA2 based upon [4].

A. *Overview of* VIATRA2 *transformation language*

The Viatra Textual Command Language (VTCL) consists of several constructs that together form an expressive language for developing both model to model transformations and code generators. Graph patterns (GP) define constraints and conditions on models, graph transformation (GT) [5] rules support the definition of elementary model manipulations, while abstract state machine (ASM) [6] rules can be used for the description of control structures.

Graph patterns, negative patterns Graph patterns (Fig. 1) are the atomic units of model transformations. They represent conditions (or constraints) that have to be fulfilled by a part of the model space in order to execute some manipulation steps on the model. Patterns may have parameters listed after the pattern name. The basic pattern body contains model element and relationship definitions defined by VTML constructs ([7]).

```
// Describes a Place and its Token instance.
pattern placeToken(P, T) =
{
// variable P is an instance of 'Place'
'PetriNet'.'Place'(P);
// variable T is an instance of 'Transition'
'PetriNet'.'Transition'(T);
 // they connected by a 'tokens' relation
'PetriNet'.'Place'.'tokens'(R, P, T);
}
```

Figure 1: Graph pattern: a Place and its Token.

A model (i.e. part of the model space) can satisfy a graph pattern, if the pattern can be matched to a subgraph of the model using a generalized *graph pattern matching* technique presented.

In VTCL, *patterns may call other patterns* using the *find* keyword. This feature enables the reuse of existing patterns as a part of a new (more complex) one.

```
// Removes a token from the place 'Place'.
gtrule removeToken(in Place) =
{
 precondition find placeToken(Place, Token)
 postcondition find place(Place)
}
```

Figure 2: GT rule: removing a Token from a Place.

Graph transformation rules Graph transformation (GT) [5] provides a high-level rule and pattern-based manipulation language for graph models. In VTCL, graph transformation rules (Fig. 2) may be specified by using a *precondition* (or left-hand side – LHS) pattern determining the applicability of the rule, and a *postcondition* pattern (or right-hand side – RHS) which declaratively specifies the result model after rule application. Elements that are present only in (the image of) the LHS are deleted, elements that are present only in RHS are created, and other model elements remain unchanged. The LHS and RHS patterns share information on matchings by parameter passing.

Complex transformation programs To execute graph transformation rules, they have to be invoked from a transformation program. In this case, the actual parameter list of the transformation has to contain a valid value for all input parameters, and an unbound variable for all output parameters. A rule can be executed for all possible matches by quantifying some of the input parameters using the *forall* construct.

```
rule fireTransition(in Tr) = seq {
  forall P with find sourcePlace(Tr, P) do
      apply removeToken(P);
  forall P with find targetPlace(Tr, P) do
      apply addToken(P);
}
```

Figure 3: MT rule: firing a Transition.

To illustrate how a VIATRA2 transformation program can be used to execute a simulation step, a small VIATRA2 transformation program is shown on Figure 3. It takes a Transition instance as input, locates all Places which are connected to the Transition by an OutArc (sourcePlace pattern), removes a token from each of those Places (removeToken rule), locates all Places which are connected to the Transition by an InArc (targetPlace pattern), and adds a token to each of those Places (addToken rule). In this case, the `forall` ASM construct is used to compute all matches of the sourcePlace and targetPlace patterns, and two simple graph transformation rules are applied to facilitate the removal and addition of Tokens.

Enabledness calculation A graph pattern expressing the enabledness condition for the simple fire simulation step can be used with the `forall` construct, which generates all matchings for that pattern.

IV. User-guided transformations for model simulation

To facilitate design-time interactive model execution, the ViatraDSM framework provides domain-specific user-guided simulation built on top of VIATRA2 model transformations. The fundamental idea is that each atomic model simulation step (modifying the instance model but preserving syntactic validity) is a VIATRA2 model transformation, defined over the abstract syntax metamodel of the domain. User interaction may be provided when the execution of step is fully finished. The user may continuously observe the changes of the domain-specific model as the simulation steps are executed, and provide input at non-deterministic choice points. This interaction is specified by a state transition system based on guarded commands.

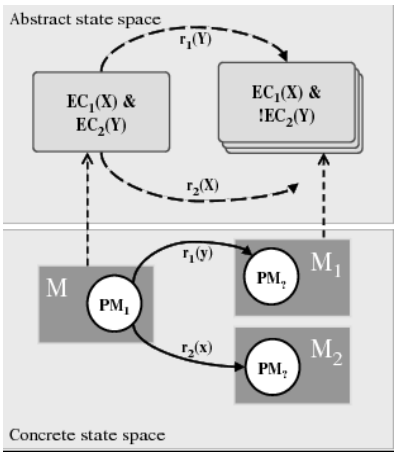


Figure 4: Abstract and concrete state spaces.

A. *The simulation process*

An abstract simulation execution, which is perceived by the user, can be formalised using a state transition system specified by guarded commands. Transformation programs defined in Sec. III. can be grouped into two categories: (i) *enabledness conditions* EC and (ii) *simulation rules* R. Simulation rules can be used to manage control and data flow; it is up to the designer to separate these concerns if the domain necessitates. Enabledness conditions act as guards for the execution of simulation rules. The simulation process is illustrated in Fig. 4 using abstract (top) and concrete (bottom) steps.

The abstract system evolves along transitions by executing a simulation rule r provided that the guard of this rule g_r is satisfied. This abstract guard is a logical formula composed of (positive or negative) enabledness conditions as literals. The abstract transition system captures the high-level view of a simulation as perceived by the designer.

All transitions on the abstract level are *potential* transitions, i.e. there may but not necessarily be a corresponding concrete transition in the system model. For instance, if the ”fire transition” simulation step rule is be fireable for transition T1, but not fireable for transitions T2 and T3, this still becomes a valid transition on the abstract level. However, due to the dynamic nature of the system state, it is frequently infeasible to a priori calculate all the transitions on the abstract level. Instead, the execution of an abstract transition is derived from executing steps on the concrete level as follows:

- First all enabledness conditions (e.g. $EC_1(X)$ and $EC_2(Y)$ in Fig. 4) are evaluated by initiating graph pattern matching on the current concrete state M in order to determine the current abstract state. An enabledness condition is satisfied if there is at least one match on the concrete level. The validity of enabledness conditions determine the abstract state, e.g. if $PM_1(x, y)$ is a part of the system model which satisfies both $EC_1(X)$ and $EC_2(Y)$ along some match $X = x$ and $Y = y$, then one can conclude that the abstract state machine is in the abstract state $EC_1(X) \wedge EC_2(Y)$.
- Then we check on the abstract level, which simulation transitions are enabled, i.e. lead out from the current abstract state (e.g. $r_1(Y)$ and $r_2(X)$ in Fig. 4. Exactly these enabled simulation rules will be offered to the designer for user interaction. If there are multiple matches of a simulation rule on the concrete level, they denote additional user choice points. This way, a user can select the next simulation rule to apply as well as the concrete matches (e.g. $Y = y$) where the corresponding rules are applicable.
- After the user makes his or her choice, the simulation rule is executed for the given match on the concrete level to derive a new model M_1 from M as a result of the atomic simulation step. After this, the process starts over again by evaluating the enabledness conditions in M_1.

B. Evaluation of enabledness conditions

While enabledness conditions are most typically expressed using a graph pattern, in some cases it may be necessary to perform more complex computations. Therefore, the ViatraDSM framework uses normal VIATRA2 transformation programs for this purpose as well: "check machines" traverse the model space and produce appropriate output marking for those elements which may be used as execution inputs for the given simulation step.

Since model space traversal is required at each choice point, our technique is best suited for domains where most simulation steps are limited in scope (*locality* principle) – which is the case with many executable modeling domains. In addition, we are developing a technique to speed up the evaluation of enabledness conditions by *incremental pattern matching* [8]. In this approach, graph patterns only need (partial) re-evaluation after the model has changed, since underlying mechanisms automatically track model space alterations and update the bindings of a previously found match accordingly, which may result in a significant speed up.

V. Conclusions and Future Work

In the current paper, we presented how interactive design-time model simulation can be specified on a high-level of abstraction (i) by the transformation language of VIATRA2 for capturing atomic model simulation steps, and (ii) by a state transition system for modeling user interaction, as integrated in the ViatraDSM framework.

While none of the core techniques (i.e. using model transformations as atomic steps, statemachines to describe user interaction) are unprecedented separately in themselves, our combination is a significant contribution both in DSML frameworks and in the field of model transformation where no support is existent for user-guided model transformations driven at a high-level of abstraction.

In the future, we primarily aim at developing support for incremental model transformations [8], which can significantly speed up model simulation. In this approach, matches are stored explicitly, and refreshed incrementally when the underlying model is changed. As a consequence, we expect a drastic reduction in the evaluation of enabledness conditions.

References

[1] The Eclipse Project, "Graphical Modeling Framework," http://www.eclipse.org/gmf.

[2] Microsoft, "DSL Tools," http://lab.msdn.microsoft.com/teamsystem/workshop/dsltools/default.aspx.

[3] I. Ráth and D. Varró, "Challenges for advanced domain-specific modeling frameworks," Nantes, France, July 2006.

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

[5] 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.

[6] E. Börger and R. Särk, *Abstract State Machines. A method for High-Level System Design and Analysis*, Springer-Verlag, 2003.

[7] D. Varró and A. Pataricza, "VPM: A visual, precise and multilevel metamodeling framework for describing mathematical domains and UML," *Journal of Software and Systems Modeling*, 2(3):187–210, October 2003.

[8] G. Varró, D. Varró, A. Schürr, "Incremental graph pattern matching: Data structures and initial experiments," in *Proceedings of the 2nd International Workshop on Graph and Model Transformation*, Brighton, United Kingdom, September 2006.

MODEL CHECKING BASED VERIFICATION OF UML 2.0 STATECHARTS

Áron SISAK
Advisor: István MAJZIK

I. Introduction

This work aims to transform UML 2.0 statechart models with their formally specified semantics to a model checking language suitable for formal verification. The transformation approach allows to perform an early investigation of software models in a very rich modeling language.

II. UML 2.0 Statecharts

UML 2.0 Statecharts is the de facto standard formalism for modeling the behavior of reactive, state-based systems. Its graphical representation allows visual system modeling, while its semantics aims to serve as a base for both generating code and reasoning about the properties of the system being modeled. The language is very rich, e.g., it allows capturing the hierarchical refinement of models, it makes possible to employ guards and to define compound transition and activity structures.

Besides the powerful abilities defined, the UML 2.0 standard contains certain ambiguities as well. The thesis of Gergely Pintér [1] offered remedy for these problems with a formally defined precise statechart semantics. Furthermore, the semantics is defined in a *declarative* manner that suits the declarative model checking paradigm very well. This work is based upon this semantics, called Precise Statecharts.

III. Formal Verification

Formal verification has the goal to prove that an examined system satisfies certain properties, based on an underlying rigorous mathematical model. Among various formal verification approaches I concentrated on model checking, mostly due to its automated nature.

A. Model Checking

Model checking assumes a formal description of the system (i.e., a system model) and performs an exhaustive exploration of the model to verify the desired properties. In other words, the user defines the model of the system, and the requirements (properties, theorems) and the model checker explores all possible execution traces of the model and checks if the requirements hold on all of them.

B. SAL tools

The SAL model checking framework [2] is used to perform the actual model checking. *SAL* stands both for a model checking program suite and its model specification language. The model checking suite includes a "traditional" BDD-based and a novel "bounded" model checker, a simulator, a tool to check for deadlocks, etc.

The language is based on the Kripke Transition System formalism, containing states, transitions and state variables. The latter can be built up from primitive types via the rich type system in SAL [3]. More information about SAL model checkers and its specification language is available at [4].

IV. Precise Statecharts to SAL Model Transformation

The implementation of this model transformation task is based on the Java API being developed for the Precise Statechart Semantics. This API makes possible to process UML2 statechart models produced by some EMF-based UML modeling tool, with access to the special constructs defined by the precise semantics (mostly sets and relations).

Although SAL has multiple equivalent syntaxes, the ASCII variant is preferred because it is suitable both for code generation and for humans willing to read and eventually complete the code. I created a prototype template-based solution, using the Velocity template engine [5]. This way metamodel level constructs can be implemented without any knowledge about the actual model being processed during the transformation (e.g., initializing a statechart or firing a transition).

V. Model Transformation Implementation Details

The data structure and the generic behavior of a statechart (i.e., statechart semantics) is defined using the Velocity template with the statechart semantics captured by the transitions of a SAL module. The states, regions, triggers and other statechart elements are implemented as enumerations, generated from the actual statechart model. The guards and effects are implemented as parameterized generic code, copied from the model if they are available. The generator uses the events and call behavior implementations from the actual model using the template. The actual run-time data consists of a set of boolean arrays representing the active states in the statechart structure, and a set of SAL variables used in the guards and activities.

The most difficult aspects of the transformation are the following: *1)* enabling multiple activities during compound transitions, which are defined in a general way using PERT graphs in the Precise Statecharts semantics; *2)* implementing the use of parallelism (i.e., fork and join constructs); *3)* and modeling an event queue, especially capturing the possibility of adding events into the event queue from activities. It is also considerably difficult to use the proper SAL constructs to capture semantics possibly avoiding state space explosion, e.g., using simple boolean arrays instead of records can reduce the state space with multiple orders of magnitude.

VI. Conclusion and Future Work

I created an initial prototype of the SAL model generator from UML2 statecharts. For simplicity reasons it transforms a single statechart, but the approach can be extended to support multiple ones. There is a limited support for guards and activities: external statechart variables are modeled explicitly and arbitrary (hand-crafted) SAL functions can be used on these to specify guards and activities.

Handling variables has room for improvement: several transformation issues can be automated, most notably deriving guards and effects directly from the source model. Event queue handling and support for the most advanced compound transition structures are quite complex tasks with very limited support now; both are to be improved in the near future. The prototype engine also needs extensive testing with various input models and requirements (i.e., properties), possibly from different applications domains.

References

[1] G. Pintér, *Model Based Program Synthesis and Runtime Error Detection for Dependable Embedded Systems*, Ph.D. thesis, BME, Department of Measurement and Information Systems, 2007.

[2] L. de Moura, S. Owre, and et al., "SAL 2," in *Computer-Aided Verification, CAV 2004*, R. Alur and D. Peled, Eds., vol. 3114 of *Lecture Notes in Computer Science*, pp. 496–500, Boston, MA, July 2004. Springer-Verlag.

[3] L. de Moura, S. Owre, and N. Shankar, "The SAL Language Manual," CSL technical report, SRI International, 2003.

[4] "The SAL homepage," `http://sal.csl.sri.com/`.

[5] "The Velocity homepage," `http://velocity.apache.org/`.

MODEL BASED EVALUATION OF ACCESS CONTROL

Dániel TÓTH
Advisor: András PATARICZA

I. Introduction

Access control is a factor with especially high associated risk in infrastructures where the initial design usually do not take security constraints into account and access control is developed in an evolutionary manner. In such systems further applications are frequently added to the pool of programs acting as the functional blocks of a large scale integrated application without any supervision of the potential dangers and risks of using outdated access control schemes.

Recently both regulatory compliance requirements and the growing need of enterprises to comply to practical system management standards like ITIL, have started enforcing certain access control policy properties over data and their access methods.

The existing formal requirement systems such as Bell-LaPadula [1] for confidentiality, Biba [2], Clark-Wilson[3], or the recently published Shankar-Jaeger-Sailer [4] can be incorporated into high level security policies of an organization. Such formalism can enable automated security auditing, however this is an emerging problem that needs new approaches and tools.

II. Practical challenges of Access Control evaluation

A. Heterogenity

The evolutionary development of the software used in IT infrastructures has lead to the existence of various different, often incompatible access control schemes. Large scale enterprise infrastructures are typically heterogeneous in the software involved. This problem manifests as the heterogenity of operating system as well as the interaction between the application and operating system level access controls. Clearly any comprehensive means of analysis must treat the entire deployment as a whole, otherwise the disregard of possible interactions could result in missed access paths.

B. Scalability

The size of a typical enterprise deployment in the number of both the involved computer systems and the users presents another challenge for comprehensive analysis. The size of the Access Control Matrix is proportional to the number of protected objects by the number of actors. It can be seen that traditional formal analysis algorythms that rely on computing the access matrices become unfeasible when dealing with a large complex deployment.

III. Model-based approach

In this article I propose a model-based approach for the development of access control evaluation tools that deal with heterogeneous systems as well as some techniques that can improve the handling of large data sets.

The basic concepts of OMG's Model Driven Architecture (formulating the requirements over an abstract platform-independent metamodel (PIM) while the exact implementation details specific to a target platform are represented by instances of platform-specific metamodels (PSM)) are key elements of this approach.

However, MDA's top-down design process has to be substituted with a bottom-up procedure in the context of analyzing existing systems, simply, because the existing systems are not engineered with

the model-based concept in mind, moreover they were subject to evolutionary development. Note that the following process outline is generally applicable to any kind of consolidation problem, not just for access controls:

1. Identitfy concepts and evaluation semantics of existing systems to formulate platform specific metamodels.
2. Synthetize the platform metamodels into a common Platform Independent Metamodel.
3. Implement a means of discovery for instantiating a platform specific model precisely representing the actual deployment.
4. Implement a transformation that will abstract the discovered platform specific models into an instance of the PIM.
5. Carry out formal analysis on the PIM.

A. Analysis of existing systems

As part of this research the access control subsystems of the following operating systems were analysed and metamodelled:

- Linux with standard POSIX filesystem level access
- Linux with sudo access control
- Linux with optional ACL extensions
- Windows Server 2003 [5]
- VMWare ESX Server virtualization platform

Each platform implements a more or less complete Role-Based Access Control scheme (RBAC)[6]. This manifests in the ability to assign actors into groups or roles, thereby giving rise to actor hierarchy.

A common feature is ability to "impersonate", i.e. to change the effective user identifier during execution, ultimately operating on behalf of another actor. Each platform implements this differently. This feature is dynamic making it is particularly hard to analyse, as the internal execution flow of the applications making use of this feature needs to be traced. Static analysis can only make either optimistic or pessimistic assumptions.

Another common feature among ACL supporting platfroms is the possibility of permission inheritance, which is very important since it allows the administrator to reduce the number of necessary explicite access control definitions. Each platform has its own complex ACL evaluation algorythm for handling such implied permissions.

B. Defining the Platform Independent Metamodel

Carrying out analyses on installations of complex heterogeneous systems is only possible if the incoherently structured (instances of different metamodels) deployment information is converted to a uniform representation. There is a choice between two substantially different kinds of platform independent metamodels.

In the "vertical" approach, the PIM needs to capture only common, basic concepts (primitives) of the problem domain. This will result in a flattened model instance (similar to the access matrix formalism) that is straightforward to formally analyse but too large to handle. For instance, all implied permissions become explicitely represented in this approach.

The "horizontal" approach on the other hand contains all the platform specific metamodels as a union, thus creating a common metamodel merging the platform specific metamodels into a single notion. In this case the concrete model can be a precise equivalent of the configuration of the system component under analysis, so the exact deployment can be reproduced from the model. Obviously, this

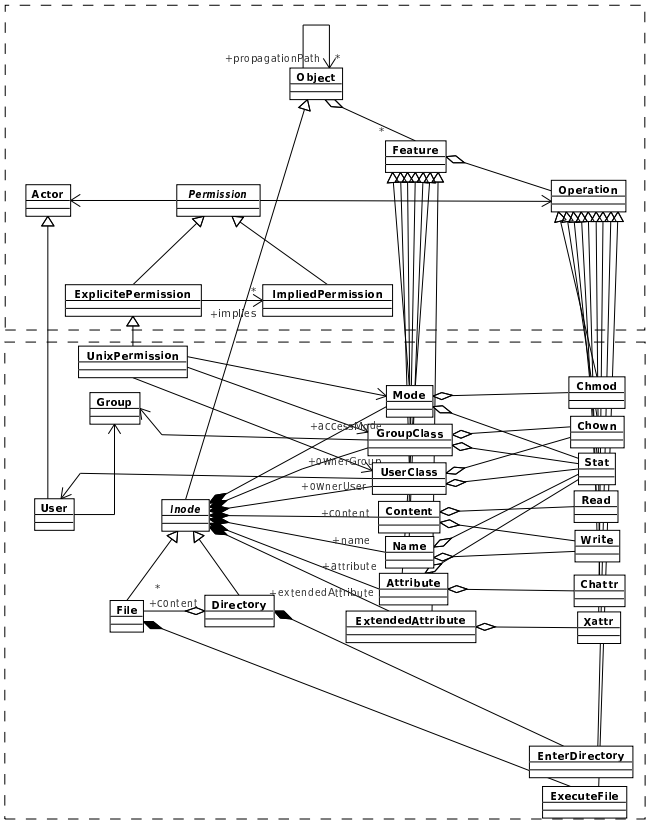


Figure 1: The platform independent metamodel (above) with POSIX platform specific metamodel as an example (below)

extreme will result in a relatively compact model instance, but any formal analysis will be troublesome to develop as the details of each platform will have to be dealt with.

The two approaches can be combined by realizing that the origin of complicated, diverse access modes (read, write, change permission, change ownership etc.) is simply the consequence of the insufficiently detailed modelling of protected objects. File system entries for example are not atomic entities but objects with separate attributes. By modeling the various attributes (including the permission attributes themselves) separately, all access modes can be mapped to simple read, write and execute operation primitives. This way a platform independent core metamodel can be devised that serves a generalization for all platform specific ones.

C. Semantic modeling

Remaining issues of the above described generalization concept are:

- Handling access to attributes that define permissions themselves
- Actor hierarchy
- Efficient handling of implied (inherited) permissions

It is desireable from the scalability standpoint to avoid explicitely instantiating the implicite permissions described. It should also be noted that all three outlined issues are similar in nature, they all describe implied permissions that can only be evaluated by simultaneously testing multiple conditions on different model elements separated by non trivial, recursive navigational patterns.

It is important to point out that each platform uses different semantics for implied permissions. Implementing such complex evaluators for heterogenous systems in traditional imperative programming languages is an error-prone task. An important part of this research is the investigation of using recursive pattern matching in addressing the above semantics. Until now, there were no specific requirements for the modeling language, however this approach proposes the use of a graph transformation

framework with efficient recursive pattern matching capabilities, such as VIATRA2 R3 [7].

This way the actual access evaluation becomes a pattern matching problem. Graph patterns can be considered as a formal way of defining the access evaluation procedure. The development process thus includes not only the platform specific metamodel definition but also the need to define a supplementary pattern library for each platform. These platform specific pattern libraries can then be combined into a generic access evaluation pattern that serves as a common interface for all platforms.

Similarly the formal requirements, such as the no read-up, no write-down rules of Bell-LaPadula and similar rules of Biba integrity model can also be defined as patterns. These patterns are also generic, not directly refering to the actual model elements but calling the generic access permission evaluation patterns instead. This way the same formal analysis becomes available for all platforms without additional effort.

IV. Proof of Concept

An early proof of concept implementation utilized IBM Tivoli CCMDB for discovering and storing the access control and identity information for Linux systems. The configuration data was then parsed and instantiated as EMF models. The platform independent core metamodel was also defined in EMF along with transformation and formal analysis code written Java. The output of the formal analysis is an audit report model that traces all possible violations of the formal requirements. Although this implementation is functional it required significant development effort to support just a single platform, which showed the drawback of this approach.

By utilizing the EMF integration facility of VIATRA, the EMF access control models were adapted to the VIATRA framework. At the time of writing, the integration of multiple target platforms is being developed using the described recursive graph matching approach.

V. Conclusion and further work

In this paper, I outlined the basic development process and key considerations for model-based evaluation of heterogeneous access control deployments. The most important direction of future work is to test the pattern matching-based implementation on actual deployments to find out its practical limits. Incorporating clustering of protected objects with similar permissions will also be investigated to bring further reduction in the resource requirements. Another important area to investigate is the recently developed incremental pattern matching facility of VIATRA2. Finally implementing more fine-grained formal analyses such Clark-Wilson are considered for future work. Such formal requirement systems involve the opertional analysis of the applications which greatly broadens the scope of this research.

References

[1] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," *MITRE Technical Report 2547*, I., Mar. 1973.

[2] K. J. Biba, "Integrity considerations for secure computer systems," *MTR-3153, The Mitre Corporation*, Apr. 1977.

[3] D. Clark and D. Wilson, "A comparison of commercial and military computer security policies," in *IEEE Symposium on Computer Security and Privacy*, Apr. 1987.

[4] T. J. U. Shankar and R. Sailer, "Toward automated information-flow integrity verification for security-critical applications," in *2006 Network and Distributed Systems Security Symposium*, pp. 267–280, 2006.

[5] *Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server 2003, Windows XP and Windows2000*.

[6] R. K. R. Sandhu, D. Ferraiolo, "The nist model for role-based access control: Towards a unified standard," in *5th ACM Workshop on Role-Based Access Control*, pp. 47–63. ACM Press, 2000.

[7] G. V. Á. Horváth and D. Varró, "Generic search plans for matching advanced graph patterns," in *Sixth International Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT 2007)*, pp. 57–68, 2007.

SOME PRACTICAL CONSIDERATIONS ON USING SVMS AND OTHER KERNEL-BASED METHODS

Sékou Tidiani COULIBALY
Advisor: Gábor HORVÁTH

I. Introduction

Statistical learning theory opened the era of kernel methods, and many algorithms related to classical pattern recognition are being rewritten (extended) to their kernel formulations. This paper puts emphasis on practical considerations that may have influence on the performance goal of SVMs or any other kernel based procedure, i.e. maximization of generalization ability and capacity control.

Unlike multilayer neural network support vector machines use direct functions to implement its goals. In two-class, $x_i \in \Re^n$ instances, classification problems and in the linearly separability case with M training samples $(x_{i,} y_i) \in X \times \{\pm 1\}$, we use $W^T x_i + b \geq 1$ for $y_i = 1$ and $W^T x_i + b \leq -1$ for $y_i = -1$ decision functions. The hard margin SVM implementing the optimal separating hyperplane, having W as orthogonal vector, can be obtained by solving the quadratic problem given by Eq.(1).

$$Q(W, b, \alpha) = \tfrac{1}{2} W^T W - \sum_{i=1}^{M} \alpha_i \{ y_i (W^T x_i + b) - 1 \} . \quad (1)$$

In Eq.(1), α_i associated with each training sample are the nonnegative Lagrange multipliers.

Many problems of practical interest fall in the case of nonlinearly separability, then Eq.(1) has no feasible solution and the hard-margin SVM is unsolvable. The separating hyperplane is then constructed following the principles of structural risk minimization (SRM) [1]. The algorithm leads to the quadratic optimization problem as shown below:

$$Q(W, b, \alpha, \xi) = \tfrac{1}{2} W^T W + \tfrac{1}{2} C \sum_{i=1}^{M} \xi_i^p - \sum_{i=1}^{M} \alpha_i \{ y_i (W^T g(x_i) + b) - 1 + \xi_i \} . \quad (2)$$

The optimal solution must satisfy either α_i =0 or condition of Eq.(3).

$$y_i \{ \sum_{j=1}^{M} \alpha_j y_j (K(x_j, x_i) + \delta_{ij} . \frac{1}{C}) + b \} - 1 = 0 . \quad (3)$$

Here, δ_{ij} stands for Kronecker's delta function; it is 1 for $i = j$, otherwise 0. In Eq.(2), $g(x)$ is typically a nonlinear mapping function to the feature space; ξ the slack variables gives chance of having feasible solution to the optimization problem. When p value takes 1 or 2 we have L1 SVM or L2 SVM, with the former having the associated computations stable and an enhanced generalization [6]. From the above equation it can be seen that to construct an efficient support vector machine one can use a positive definite function $K(x_j, x_i)$ called kernel function. Different types of kernel are applied to solve kernel specific problems. All the above equations show that in SVM (kernel methods) paradigm there are two major stages. The first is the regularization frame to formulate the optimization problem to solve and define the parameter(s) controlling the margin size during training the SVM. The second is the SVM model selection and the choice of the kernel function.

II. Regularization

Minimizing the "pure" empirical risk can lead to numerical instabilities and bad generalization performance. Regularisation is a possible way to avoid these problems and a procedure to restrict the class of admissible solutions, for instance to a compact set. This technique was introduced by

Tikhonov and Arsenin [1, 2] for solving inverse problems and has since been applied to learning problems hence they are generally ill-posed.

In various SVM algorithms the parameter *C* controls the trade-off between complexity and portion of data samples that are nonseparable; it determines implicitly the size of margin. This margin embeds the criterion of falsifiability [1, 4], formulated by K. Popper as necessary condition for a true theory to be falsifiable by certain observations, facts or data samples. The practice (principle) using the above idea, the SRM can be formulated as follow:

- Minimize the total empirical risk for data samples lying inside the margin.
- Achieve maximum separation (margin) between training data samples that are correctly classified.

The coefficient of the penalty term, *C*, also needs to be defined by the user. Again, there is no easy method for selecting its value aside from evaluating the resulting model's performance on a validation set.

III. Kernel function selection

The choice of kernel function is crucial in all kernel-based algorithms. The kernel function is seemed to be a prior knowledge that is available about a task in addition to the empirical observations. Although the question of how to choose the "best" kernel function for a given dataset is often posed, it has no "good" answer on top of this and it may be crucial for success.

A more formal metric for choosing the best kernel is provided by the upper bound on the VC dimension [1, 2, 3]. However, this remain an assumption even though the VC dimension describes the complexity and flexibility of the kernel, it does not provide practical proof that the chosen kernel is the "best" one. The choice can be validated by methods such as cross-validation.

Originally SVMs are based on fixed-length input data. For non-vector based applications there are two approaches to handle in SVMs or other kernel algorithms. The first is to extract vector-based features. The second is to create feature-specific kernel functions.

The list of existing kernels in literatures on the field is growing: linear, polynomial, radial basis, string, three-layer neural network, Hausdorff, and histogram intersection kernel functions [3, 4].

Beside the choice of the proper kernel function, may arise the task of kernel preconditioning. In fact, when the number of input variables is very large numerical problems make difficult the training of SVMs. Some examples of kernel preconditioning (normalizing) are given in [5], the handling of the bias term in polynomial and RBF kernels are also discussed.

IV. Conclusion

A great number of solutions to problems that is related to classical pattern recognition have adopted kernel paradigm: Kernel PCA and Kernel Feature Analysis, Regularized Principal Manifolds, estimation of the support of a distribution, Kernel Discriminant Analysis and Relevance Vector Machines [1, 3], are involved in the kernel selection problem and other here non mentioned algorithms need the development application specific kernel functions.

The knowledge about various regularization (function penalisation) techniques can be useful in solving kernel-based problems [3].

References

[1] V. N. Vapnik, "Estimation of Dependencies Based on Empirical Data", Springer, 2006

[2] R. Herbrich, "Learning Kernel Classifiers: theory and algorithms", The MIT Press Cambridge, 2002.

[3] B. Schölkopf, A. J. Smola, "Learning with Kernels: Support vector Machines, Regularization, Optimization, and Beyond", The MIT Press, 2002.

[4] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

[5] S. Abe, "Support Vector Machines for Pattern Classification", Springer-Verlag, New York, 2005.

ULTRA WIDEBAND LOW RATE COMMUNICATIONS IN EMBEDDED APPLICATIONS: A CHAOS-BASED APPROACH

Tamás KRÉBESZ
Advisor: Géza KOLUMBÁN

I. Introduction

UWB wavelets where the power of transmitted signal is spread over an extremely wide frequency band make the reuse of frequency bands already occupied by conventional narrowband radio systems possible. Low data rate wireless communication systems are required in many embedded applications where such LR-UWB system should operate in an ad-hoc network for years without maintenance and the price of one UWB device has to be less than USD10.00. Therefore CMOS technology and simple transceiver architecture has to be used in implementation. To satisfy these requirements a brand new approach is needed where new carriers, modulation schemes and transceiver architectures are used.

II. Wavelets in UWB radio

The only restriction on UWB wavelets defined by ESTI and FCC is the power spectral density (psd) of transmitted signal. It has to be kept as low as -41.3 dBm in 1-MHz bandwidth. The UWB frequency band goes from 3.1 GHz to 10.6 GHz. Since psd allowed for UWB wavelets is extremely low to get a useable radio coverage ultra-wideband wavelets must be used. The typical bandwidths of UWB signals are 500 MHz and 2 GHz. The wavelets are fixed waveforms and chaotic signals in impulse radio [1] and chaotic UWB radio [2], respectively.

III. Demodulation schemes in UWB applications

Wavelets used in UWB radio have very short duration (about 0.5 ns). Coherent demodulation cannot be implemented because the recovery of UWB wavelets is not feasible. Proposed noncoherent demodulation schemes are: (i) Pulse Polarity Modulation (PPoM) with one wavelet and template detection [1], and (ii) Transmitted Reference (TR) system with two wavelets and autocorrelation detection [2, 3].

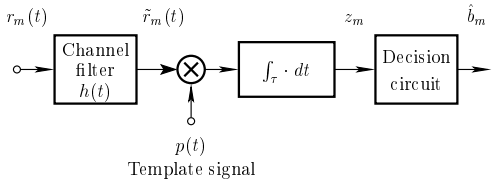


Figure 1: Detection of PPoM with a template signal $p(t)$.

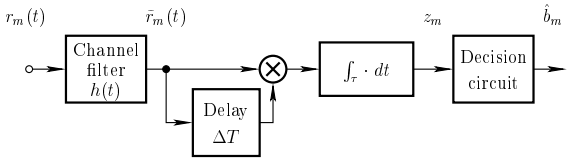


Figure 2: Block diagram of TR autocorrelation receiver.

A. PPoM with Template Detection

In PPoM one arbitrary but fixed waveform $g(t)$ is used to carry digital information. Because of its excellent spectral properties, a bell-shaped Gaussian impulse is generally used as $g(t)$[3]. Information is mapped to the sign of $g(t)$. Let $r_m(t) = g(t)+n(t)$ and $\tilde{r}_m(t) = \tilde{g}(t)+\tilde{n}(t)$ denote the received noisy signal before and after channel filtering, respectively. The information $\hat{b}_m$ is recovered by correlating $\tilde{r}_m(t)$ with a template signal $p(t)$ as shown in Fig. 1 where z_m is the observation signal.

The template signal is a windowing function

$$p(t) = \begin{cases} \frac{1}{\sqrt{\tau}}, & \text{if } |t| < \frac{\tau}{2} \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

where τ is the observation time period.

The drawback of template detection is that the demodulator is very sensitive to errors in window width (see Fig. 3) and timing. Any error reduces the separation of message points in the observation space and results in a considerable performance degradation.

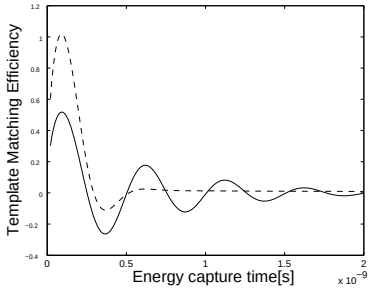


Figure 3: Template matching efficiency as a function of the width of template signal. RF bandwidths of Gaussian impulses are 4 GHz (dashed curve) and 1 GHz (solid curve).

B. Transmitted Reference (TR) system

In TR system two wavelets, called chips, are used to transmit one bit information [2, 3]. The first chip serves as a reference, while the second one carries the information. Bit 1 is sent by transmitting the chip twice. For bit 0, the reference chip is transmitted, followed by an inverted copy of the same signal. The reference chip also serves as a test signal used to measure the actual channel characteristics.

Due to the special structure of TR signal, information bits may be recovered from the sign of the correlation between the reference and information bearing chips as shown in Fig. 2.

Noise performances of template detector and TR system [2] can be seen in Figs. 4 and 5, respectively. According to the BER curves the two systems are very similar in their noise performances.

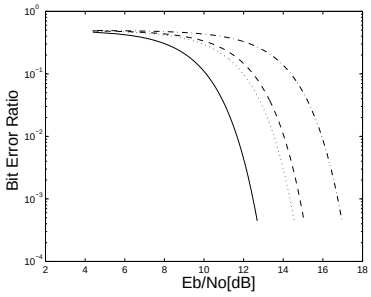


Figure 4: Noise performance of pulse polarity modulation built with template detection.

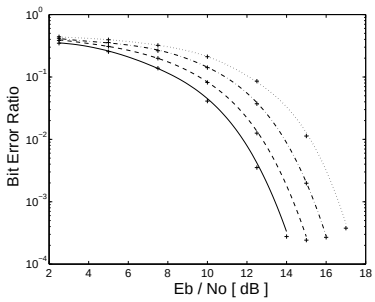


Figure 5: Noise performance of a TR system built with autocorrelation receiver.

IV. Conclusions

The main advantage of template detection is that template signal is a noise-free signal, its application results in a better noise performance if the template is perfectly matched to the UWB wavelet. However, even a small timing error corrupts the noise performance considerably, it may even prevent the communications.

FM-DCSK [2] and the TR modulation scheme offer an alternative noncoherent UWB modulation scheme that may be used with either chaotic or deterministic wavelets. TR system is very robust against the timing error and a TR signal may be demodulated by a simple autocorrelation receiver.

References

[1] K. Siwiak and D. McKeown, *Ultra-Wideband Radio Technology*, Wiley, Chichester, UK, 2004.

[2] 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.

[3] M. Ghavani, L. B. Michael, and R. Kohno, *Ultra-Wideband Signals & Systems in Communication Engineering*, Wiley, 2nd edition, 2006.

DYNAMIC RECONFIGURATION OF FPGA DEVICES

Tamás RAIKOVICH
Advisor: Béla FEHÉR

I. Introduction

Nowadays, systems built with FPGA devices are becoming more and more widely used. Their great advantage is their flexibility that arises from their programmable nature as compared to systems using application specific integrated circuits (ASICs). In addition to conventional applications, the modern programmable logic devices appear also in the scope of high-performance computational structures. The reconfigurable, application specific computational architecture and the availability of large quantity of resources enable to achieve much better efficiency compared to the conventional solutions. When general purpose processors and reconfigurable computational structures are used together, the most advantageous properties of both components can be utilized. Algorithms (searching, sorting, signal and image processing, etc.) that are implemented such way have achieved 100 times or 1000 times better performance.

Using reconfiguration, designers can dramatically increase the functionality of a single FPGA, allowing a system to be implemented with fewer and smaller devices than is otherwise required.

This paper introduces a simple reconfigurable system and focuses on the Xilinx Virtex FPGA families and Xilinx development software.

II. Configuration of the FPGA devices

The FPGA devices provide a number of serial and parallel configuration interfaces that can be used to download the configuration data to the device. During the normal configuration process, the operation of the FPGA is suspended, the existing configuration is cleared, the new configuration is loaded into the configuration memory and then the device is restarted.

There are FPGA devices that support partial dynamic reconfiguration. During the partial dynamic configuration process, the device remains fully functional while a given part of the FPGA is reconfigured.

The Xilinx Virtex-II, Virtex-II Pro, Virtex-4 and Virtex-5 FPGAs have a special configuration interface called ICAP (Internal Configuration Access Port). Using this internal hardware module, the downloaded application itself can reconfigure a part of the FPGA device [1].

III. A simple reconfigurable system

This design is implemented on the Avnet Xilinx Virtex-4 FX Evaluation Kit. The board contains a Xilinx XC4VFX12-FF668 (Virtex-4 FX) FPGA and it has UART and 10/100 Mbps Ethernet communication interfaces. Because of its higher speed, the Ethernet interface is used to communicate with the PC.

Figure 1 shows the block diagram of the design. Although the Virtex-4 FX devices contain embedded PowerPC processor, still the MicroBlaze soft processor was chosen. This way, the design can be easily ported to the high-performance Virtex-5 FPGA devices, which don't have embedded PowerPC processors. The LMB ICAP peripheral connects the ICAP to the data side LMB (Local Memory Bus) of the MicroBlaze processor. The LMB was chosen here because the write and read cycles on the LMB are one clock cycle length. The OPB EthernetLite is necessary for the Ethernet communication. The OPB timer has two timer registers. One timer is required by the TCP/IP

protocol, the other timer is used to measure the reconfiguration duration. The system clock frequency is 100 MHz, which is the maximum allowed clock frequency of the ICAP.

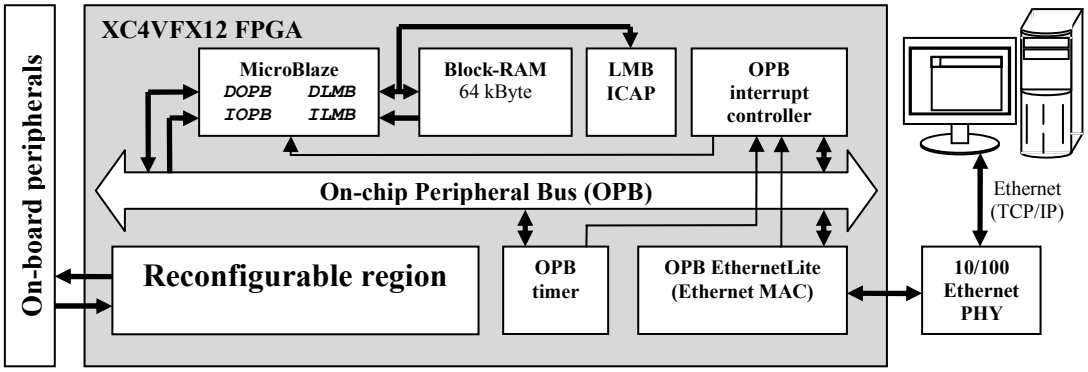


Figure 1: The block diagram of the reconfigurable system

Like normal systems, partially reconfigurable systems are also written in VHDL or Verilog hardware description language. However, these kinds of designs require much more development effort and special considerations [2]. A partially reconfigurable system consists of three main parts: the static module, the partially reconfigurable modules and the top module. Each module must be implemented in a separate project and the final merge step will generate the full and partial configuration files. The static module contains those parts of the design that won't be changed during the reconfiguration process. The partially reconfigurable modules contain those parts of the design that can be replaced with each other. All partially reconfigurable modules for a given partially reconfigurable region must be pin compatible with each other. Connections between the partially reconfigurable modules and the base design must go through a bus macro, which locks the routing between the partially reconfigurable modules and the base design. The size of the partially reconfigurable regions is determined by the minimal reconfigurable area of the given FPGA device. Lower-level modules cannot contain any clock or reset related FPGA primitives (for example BUFG, DCM or STARTUP_VIRTEX4) or I/O buffers. The top module must only contain the black-box instantiation of the lower-level modules and it also instantiates the bus macros. Xilinx ISE 9.1.02i_PR2, Xilinx EDK 9.1 and Xilinx PlanAhead 9.2 softwares were used to implement this hardware.

The software for the MicroBlaze processor uses the uIP 1.0 TCP/IP stack. As for the PC side, a simple command-line application is available to partially reconfigure the FPGA.

A future application of this design is to create a high-performance reconfigurable search engine, which performs searches in large databases. Such algorithm is used for example in medicine research where large number of molecule descriptors have to be compared with each other.

IV. Results

In this design, the partial configuration bitstream size is 23508 bytes. Using the LMB ICAP peripheral, the reconfiguration duration was 0.47 ms at 100 MHz system clock frequency. The full configuration bitstream size of the XC4VFX12 device is 595696 bytes. If the reconfiguration of the whole FPGA was assumed, the reconfiguration process would take 12 milliseconds. Consequently, an algorithm can be inefficient, if it requires many reconfigurations.

References

[1] Virtex-4 Configuration Guide, Xilinx, Inc.
URL: http://www.xilinx.com/support/documentation/user_guides/ug071.pdf

[2] Early Access Partial Reconfiguration User Guide, Xilinx, Inc.
URL: http://www.xilinx.com/support/prealounge/protected/docs/ug208.pdf

BAYESIAN ANALYSIS OF RELEVANCE IN PRESENCE OF MISSING AND ERRONEOUS DATA

Gábor HULLÁM
Advisors: Péter ANTAL, György STRAUSZ

I. Introduction

As recent technology and newly developed methods enable to address larger, more complex tasks in several different domains from telecommunication to biomedicine, the problem of focusing on the most relevant information has become increasingly important. This task has received significant attention in AI research and became known as the feature subset selection problem (FSS). In most of the cases, identifying a relevant subset of features (input variables) poses a statistical and computational challenge due to the enormous number of potential variables and to the limitations of parametric statistical methods for detecting cause-effect relationships. Motivated by this problem, we proposed a new structural model feature (property) called *Markov Blanket Graph* (MBG) in the Bayesian network framework and we formulated the *most probable MBG* (MP-MBG) problem. We demonstrated its applicability in a biomedical case study. Our current work aims to extend our present framework to handle missing and erroneous data in an integrated way.

II. The issue of relevance

The concept of relevance plays an important role in most scientific studies. Furthermore, the selection of relevant features is a central issue in several diverse fields such as information retrieval or pattern recognition. Throughout the paper we investigate this issue in the field of biomedicine.

In a typical biomedical study hundreds of clinical or genomic factors are measured in each patient sample, thus producing a dataset with a vast number of variables while the number of samples is usually moderate in comparison. In addition, the number of variables grows even further when environmental factors are added. The goal of such studies is to select the relevant features (clinical, genomic or environmental factors) w.r.t. a certain disease or condition. According to John et al.[5], given a set of features $X_i \in \boldsymbol{S}$, a target feature Y, feature relevance can be defined in the following way:

Definition 1: A feature X_i is *strongly relevant*, if there exists some x_i, y and $\boldsymbol{s_i} = x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ for which $p(x_i, \boldsymbol{s_i}) > 0$ such that $p(y \mid x_i, \boldsymbol{s_i}) \neq p(y|\boldsymbol{s_i})$. A feature X_i is *weakly relevant*, if it is not strongly relevant, and there exists a subset of features $\boldsymbol{S_i'} \in \boldsymbol{S_i}$ for which there exists some x_i, y and $\boldsymbol{s_i'}$ for which $p(x_i, \boldsymbol{s_i'}) > 0$ such that $p(y \mid x_i, \boldsymbol{s_i'}) \neq p(y| \boldsymbol{s_i'})$. A feature is *relevant*, if it is either weakly or strongly relevant; otherwise it is irrelevant.

Current solutions for this well known FSS problem include linear and logistic regression models, SVMs, PCA, likelihood based score tests, structured association methods and even ANOVA. For a detailed overview on the applicable statistical methods see [2]. A common theme in these methods is that they are based upon some form of conditional modeling. There is however another approach that can provide a solution for the FSS problem: the application of Bayesian methods.

III. Bayesian relevance analysis

The Bayesian approach has two main advantages over the others. First, a priori knowledge can be incorporated to aid the feature selection. In case that previous experiments or studies produced valuable information e.g. on clinical or genomic variable correlations, then it would be a waste to

ignore these results, since they have the potential to improve the process of selecting relevant features.

Second, the Bayesian approach enables the handling of missing values in an integrated way, while the other methods based on conditional modeling require an additional (imputation) model to resolve that problem. Since the ratio of missing and erroneous values tends to be significant, especially in case of biomedical data, appropriate handling is a central issue. One of the frequent reasons for missing values is that not all factors are analyzed in each sample due to insufficient resources or to improper study protocol. Another typical reason is that the specific value of a certain factor can not be identified unambiguously. In that case the result is either marked as erroneous with a special failed measurement symbol in the result dataset or it is completely missing.

In order to find the relevant features we used our formerly devised Bayesian relevance analysis method [1] based on Markov Blanket Graphs.

Definition 2: A subgraph of Bayesian network structure G is called the *Markov Blanket* (sub)*Graph* MBG(X_i,G) of variable X_i if it includes the nodes of MB(X_i, G) and the incoming edges into X_i and into its children, where MB(X_i, G) denotes the Markov blanket of X_i , i.e. the set of parents, children and the children's other parents for X_i under the implicit assumption that the joint probability distribution p is Markov compatible with G and stable [8].

The advantage of identifying the MBG(Y, G) of a central variable Y is twofold. First, according to definition 1, MBG(Y, G) contains all the variables X_i that are relevant w.r.t. Y. Second, it also contains the dependency relationships between these relevant variables. In terms of biomedical data analysis, when the variable Y denotes the presence or a state of the complex, multifactorial disease under study, the MBG(Y, G) contains a set of relevant factors and their interdependencies that influence that disease mechanism.

However, in case of real-world biomedical applications, the amount of data available for analysis is typically not sufficient to select a single best feature subset, i.e. a single best MBG. The *Most Probable Features* (MPFs) method described in [7] aims to resolve this matter by selecting a predefined K number of feature values with high posteriors, which minimize a given loss function. The *MP-MBG* problem is a specific case of MPFs, with the goal of finding the K most probable MBGs. By selecting the K most probable MBGs the K most probable relevant feature subsets and their interdependencies are identified. In order to facilitate this task we devised an *ordering-based estimation and search method* for Markov Blanket sets and Markov Blanket subGraphs using Markov Chain Monte Carlo sampling and the concept of MBG space as core elements [1].

Since the MBG based relevance analysis requires a completed dataset, the missing and erroneous cases must be handled in some way. The following sections describe possible approaches and methods to solve that problem.

IV. Missing value handling methods

Missing values can be classified into the following three classes [3]:

- **Missing Completely at Random (MCAR):** the probability of a missing value for a variable of interest Y is the same for all units in the population. This means that the probability of missing does not depend on either auxiliary variables X or the variable of interest Y
- **Missing at Random (MAR):** the probability of a missing value for a variable of interest Y is related to auxiliary variable(s) X
- **Not Missing at Random (NMAR):** the probability of a missing value for a variable of interest Y is related to Y or to other variables that were not observed

The correct assessment of missing value types in a dataset is crucial, because the possible ways of handling of missing values depend on it. In the following sections we discuss the most popular methods found in the literature.

A. *Complete cases method*

This is the simplest of all methods, where only the complete records are used for analysis. All other samples with at least one value missing are eliminated. If missing is unrelated to all the variables of interest (MCAR), then this method is free of bias. Even then, by rejecting incomplete samples, valuable information contained within the non-missing values is not utilized. In addition, the elimination of incomplete records may result in an unacceptably low sample size.

B. *Ad-hoc methods*

The methods in this group manipulate the variables in some way. One of the basic approaches is to transform the variable with missing values by recoding. The down side of this simple method is the considerable bias it produces. Another possible action is the elimination of the variable from the analysis. The main risk of this method is obvious, leaving a variable out means losing information, furthermore it may also have a sever impact on the dependency relationship pattern of the variables.

C. *Weighting*

Weighting methods can only be used in such a rare case, when a complete model describing the probabilities of all missing values is available. Then based on the probabilities a weight can be assigned to each of the observed values, which compensate for the missing ones.

D. *Imputation*

This is the preferred way of handling MAR type missing values. The core of imputation is to create an artificial value according to some method and replace the missing value with it. This process eventually leads to a complete data file.

Currently, we are planning to embed different imputation schemes in our Bayesian relevance analysis method and to make a comparison study. In the following section we investigate the different classes of such imputation methods.

V. Imputation schemes

A. *Heuristic filtering and imputation*

These methods either use a random number or an artificial value that is independent from the domain, as imputation source. The most general forms of these methods are *random hot deck*, *mean imputation*, *regression imputation* and *k nearest neighbour imputation*. Another possible approach is to input values according to the most probable value or configuration based on the observations.

B. *Heuristic imputation with background knowledge*

If some domain knowledge is available a priori, then it can be used to aid the imputation of missing values, e.g. in the form of predefined rules.

C. *Multiple imputation*

This class consists of methods that generate plausible values for missing observations that are imputed in every possible way, thus creating multiple completed datasets which are analyzed using complete-data methods. Finally, the results are combined, which allows the uncertainty regarding the imputation to be taken into account. Note, that multiple imputation is originated from Bayesian network methods. Although the basic principle is the same, these approaches are handled separately in the literature.

D. *Bayesian approach – likelihood based methods*

Likelihood based methods have a close connection with Bayesian methods since their implementation is most likely based on Bayesian networks. Typically, the primary goal of these approaches given a target variable Y, predictor variables $\boldsymbol{X}$ and their parameters $\boldsymbol{\theta}$ is to assess the conditional distribution: $P(Y|\boldsymbol{X}, \boldsymbol{\theta})$. When some of the predictors have missing values, then these

have to be estimated in a way so that the joint probability distribution is maximized. This is typically achieved through the use of the EM (expectation-maximization) algorithm. This approach has several variants such as the structural EM and the Bayesian structural EM [3].

VI. Asthma case study

In cooperation with SOTE DGCI we participated in a genome-wide association study (GWAS) on the asthma disease, which has a complex pathological mechanism, and possibly several genes are in connection with its symptoms. The goal of the GWAS was to study the genetic variation across the human genome in order to identify genetic associations with observable traits, or the presence or absence of the disease. More specifically we were interested in identifying relevant *single nucleotide polymorphisms* (SNPs) and *haplotypes* [4] that are connected to asthma.

The analyzed dataset consisted of 760 samples (349 cases, 411 controls) with 144 genotyped SNP variables and 20 clinical factors. Regarding only SNPs, 24.46% of the dataset was missing (7.81% completely missing, 16.65% erroneous due to measurement failure). This relatively high measurement failure rate shows the limits of the currently used genotyping technology. It allows the genotyping of only certain SNPs fulfilling strict criteria. In some cases however, even when the criteria are met, the genotype measurement can be unsuccessful, typically due to the poor quality of the sample. In this study, the erroneous values were marked with a special symbol and were processed as missing data.

In order to create a completed dataset for the Bayesian relevance analysis we applied a series of missing value handling methods. As a first step, we examined each SNP variable and removed those which had a missing value rate over 50%. This improved the overall missing value rate by 10.54%. The next step was to filter the samples in a similar way, resulting in a missing rate of 12%. After these preliminary steps, on the reminder of the dataset (685 samples, 122 SNP variables) we proceeded with the most probable value imputation calculated separately for each SNP. Finally, the Bayesian relevance analysis was applied to the completed dataset, revealing multiple relevant SNPs in connection with asthma.

The applied missing value handling methods provided a completed dataset of acceptable quality. In our future works we plan to investigate and integrate further methods into our present framework.

Acknowledgement

I am grateful to the members of SOTE DGCI SNP Lab for the chance to participate in the asthma project and to Peter Antal for many valuable discussions related to this article.

References

[1] P. Antal, G. Hullám, A. Gézsi and A. Millinghoffer, ”Learning complex bayesian network features for classification”, *In Proc. of third European Workshop on Probabilistic Graphical Models*, pp. 9–16, 2006.

[2] D. J. Balding, “A tutorial on statistical methods for population association studies”, *Nature Reviews*, *Genetics*, vol.7, pp. 781-791, Nature Publishing Group, 2006.

[3] N. Friedman, ”The Bayesian structural EM algorithm”, G. F. Cooper, S. Moral, eds., *In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 129-138, Morgan Kaufmann, 1998.

[4] International HapMap Consortium, “The international HapMap Project”, *Nature*, vol. 426, pp. 789-796, NPG, 2003.

[5] G. H. John, R. Kohavi, K. Pfleger, “Irrelevant features and the subset selection problem”, In Proceedings of the 11th International Conference on Machine Learning, pp.121-129. 1994.

[6] N. J. Horton, K. P. Kleinman, “Much Ado About Nothing: A Comparison of Missing Data Methods and Software to Fit Incomplete Data Regression Models”, *The American Statistician*, Vol. 61, No. 1, pp. 79-90, ASA, 2007.

[7] A. Millinghoffer, G. Hullám and P. Antal, “On inferring the most probable sentences in bayesian logic”, *In Workshop notes on Intelligent Data Analysis in bioMedicine And Pharmacology (IDAMAP-2007)*, pp. 13–18, 2007.

[8] J. Pearl. *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000.

ALGORITHMIC IDENTIFICATION OF BRIDGE VERTICES IN COMPLEX NETWORKS

Tamás NEPUSZ
Advisors: György STRAUSZ (BUTE), Fülöp BAZSÓ (RIPNP-HAS)

I. Introduction

The theory of complex networks, an emergent field connecting diverse disciplines such as biology, computer science and sociology, has proved to be a useful framework for the analysis of a wide range of real-world phenomena including (but not limited to) epidemic spreading, the US patent system, or technological networks such as the Internet (for an overview, see [1]). The usual approach of such analysis is to decompose the system being studied into substantially identical individual components and then characterize the system by the individual components and the network of interactions. Such a representation can then be studied by a well-established tool, namely graph theory.

A particular problem in network theory that has received considerable attention during the last few years is the detection of *communities*, i.e., subsets of network components (vertices) that interact with each other more frequently than with other parts of the network. Researchers working in the field implicitly relied on the assumption that a vertex of a network being decomposed may belong to one and only one community. However, recent research showed that this assumption can prevent one from gaining important insights on the structural overlaps in the network [2]. Vertices of a network that can not be assigned uniquely to a single community are called *bridges*, and the rest of this paper will discuss an algorithmic method for identifying bridges in complex networks by employing a fuzzy community detection framework.

II. Methods

The objective of classical community detection in networks is to partition the vertex set of the graph $G(V, E)$ into c distinct subsets in a way that maximizes intra-community edge density and minimizes the number of edges between communities. Several heuristics have been proposed to reduce the time complexity at the expense of the quality of the results obtained (e.g., [3]).

For the time being, let us assume that c is given. A convenient representation of a given partition of the graph vertices is the so-called *partition matrix* $\mathbf{U} = [u_{ik}]$, where $\mathbf{U}$ has $N = |V|$ columns and c rows. u_{ik} is the membership degree of vertex k in community i, subject to the constraint that $\sum_{i=1}^{c} u_{ik} = 1$ for all $1 \leq k \leq N$. An advantage of this framework is that it allows a vertex to belong to multiple communities with well-defined numerical degrees of membership, as long as the individual membership degrees of a vertex add up to 1. Bridge vertices can then be detected by evaluating the individual membership vectors of the vertices.

Dunn and Bezdek [4] proposed a method of identifying such fuzzy clusters in multi-dimensional datasets. However, this method requires a distance function defined in the space the data belong to, rendering it unusable for networks, where the vertices are not embedded in some multi-dimensional space. Nepusz *et al* [5] recently proposed an algorithm that circumvents the need of spatial embedding while being able to find a meaningful fuzzy partition in networks.

A. Detecting fuzzy communities and bridges in complex networks

For the sake of notational simplicity, let us denote the vector of the membership degrees of vertex k with the column vector $\mathbf{u}_k = [u_{ik}]^T$ for $1 \leq k \leq N$ and $1 \leq i \leq c$. The similarity of vertices i and j is defined as the inner product of their membership vectors: $s_{ij} = \mathbf{u}_i{}^T \mathbf{u}_j$. Similarly, the similarity matrix

of a given fuzzy partition is the matrix of similarities for all vertex pairs. It follows from the definition of similarity that $\mathbf{S} = \mathbf{U}^T\mathbf{U}$. Note that $0 \leq s_{ij} \leq 1$ and $s_{ij} = 1$ if and only if both vertices belong to the same single community.

An intuitive description of a good fuzzy partition is that it makes connected vertices as similar as possible while keeps disconnected vertices dissimilar. To formalize this, we can state that the similarity should be close to 1 in the case of connected vertices and close to zero for disconnected vertices. The discrepancy between the actual and the expected similarity can then be quantified by the following goal function:

$$D_G(\mathbf{U}) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}(A_{ij} - s_{ij})^2 \tag{1}$$

where A_{ij} is 1 if and only if vertex i is adjacent to vertex j or $i = j$ and 0 otherwise. w_{ij} is an optional weight. From now on, the fuzzy community detection reduces to the minimization of $D_G(\mathbf{U})$ with respect to $\mathbf{U}$ while keeping the column sums of $\mathbf{U}$ equal to 1.

Minimization can be done by gradient-based iterative algorithms (e.g., the method of conjugate gradients) or by stochastic methods (e.g. simulated annealing or extremal optimization). It was shown in [5] that the gradients of the goal function in the subspace satisfying the constraints on $\mathbf{U}$ are as follows:

$$\frac{\partial D_G}{\partial u_{kl}} = 2\sum_{i=1}^{N}(e_{il} + e_{li})\left(\frac{1}{c} - u_{ki}\right) \tag{2}$$

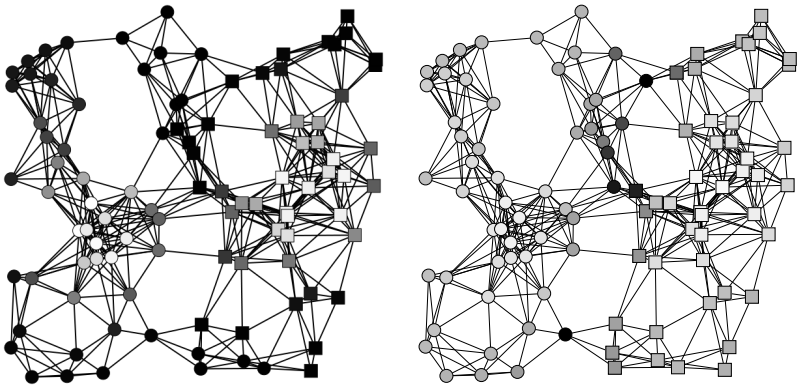
where $e_{ij} = w_{ij}(A_{ij} - s_{ij})$ (the "error term" of vertex pair i and j). The paper also provides hints on employing the gradient to find the local minima, and gives a simple heuristics that selects the number of communities appropriately. Bridge vertices in the network can then be detected by a measure called *bridgeness*. The bridgeness of vertex k (b_k) is derived from the standard deviation of the coordinates of the membership vector of vertex k. When a vertex belongs to all the communities equally, the standard deviation of the membership vector is zero (since $u_{ik} = 1/c$ for all i). The maximum deviation $\sqrt{(c-1)/c}$ is attained when the vertex belongs only to a single community. The exact definition of bridgeness follows after an appropriate renormalization:

$$b_i = 1 - \sqrt{\frac{c}{c-1}\sum_{j=1}^{c}\left(u_{ji} - \frac{1}{c}\right)^2} \tag{3}$$

B. Shortcomings of the algorithm

The method described above suffers from a minor flaw. Since the algorithm strives to make connected vertices similar and disconnected vertices dissimilar, vertices situated on the periphery of the network with only a few connections to the core will be assigned to all of the communities in order to make them as dissimilar to the rest of the networks as possible, and the few links of the vertex will not be enough to counteract this tendency. Therefore, vertices with a high bridgeness are either real bridges (structural overlaps between two communities) or outliers.

The algorithm is also slightly biased towards relatively small and densely connected communities, which becomes evident when one tries to apply the algorithm to graphs with no distinct and unique community structure. One such example is random geometric graphs [6], a special subclass of random graphs where N vertices are dropped randomly in a k-dimensional unit-sized hypercube and two vertices are connected if their distance is less than a predefined threshold ε. A random geometric graph in the unit square with $N = 100, \varepsilon = 0.2$ is shown on Fig. 1(a). The original fuzzy community detection algorithm was run with $c = 2$ and vertices are colored according to their bridgeness values. It is evident that the original algorithm detected the two "cores" of the network and classified the vertices more or less correctly, but high bridgeness values of many of the vertices is definitely an undesired effect.



(a) Original algorithm (b) Improved algorithm

Figure 1: Fuzzy communities in a random geometric graph with $N = 100$ and $\varepsilon = 0.2$ using the original (left) and the improved (right) algorithm. Vertices are colored according to their bridgeness score, darker shades meaning larger scores. Circles denote community 1 and squares denote community 2. Note that the improved method identifies real bridge vertices correctly while the original one assigns high bridgeness scores to peripheral vertices as well.

C. Improving the algorithm for better bridge identification

The improved result shown on Fig. 1(b) was achieved by taking into consideration that two vertices should be considered similar if and only if there exists an edge between them even though we wouldn't expect that edge in the graph if it were completely random. Similarly, two vertices are likely to be dissimilar if they are disconnected despite the fact that they are usually connected in a similar random graph. Formally, we consider a randomized null model of the network being studied and put emphasis on vertex pairs that show significant deviation from the null model.

A possible null model of a network is characterized by a probability matrix $\mathbf{P} = [p_{ij}]$, where p_{ij} is the probability of the existence of an edge between vertices i and j. It was shown in [3] that given a degree sequence $d_1, d_2, \ldots, d_N$ for N vertices and m edges, $p_{ij} = d_i d_j / 2m$ in the model to produce the prescribed degree sequence with maximum likelihood. The difference between the observed and the expected edge probability for vertices i and j is then expressed by $A_{ij} - p_{ij}$, and the goal function of the fuzzy clustering is modified as follows:

$$D_G(\mathbf{U}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| A_{ij} - \frac{d_i d_j}{2m} \right| (A_{ij} - s_{ij})^2 \tag{4}$$

The introduced weights can also be used for speeding up the calculation, since one can simply disregard vertex pairs with low weights, effectively reducing the number of vertex pairs that have to be considered.

III. Results

To show the effectiveness of the algorithm, we analyzed a network of weblogs about US politics and their hyperlinks. The dataset was originally published in [7] and included 1490 weblogs linked together by 19090 hyperlinks. Weblogs were annotated by their political alignment, and the vast majority of hyperlinks were placed between weblogs with similar political alignment. The dataset was converted to an undirected graph, isolated vertices and loop edges were removed and mutual links were collapsed to a single one prior to the analysis, resulting in a graph with 1222 vertices and 16714 edges. Vertex

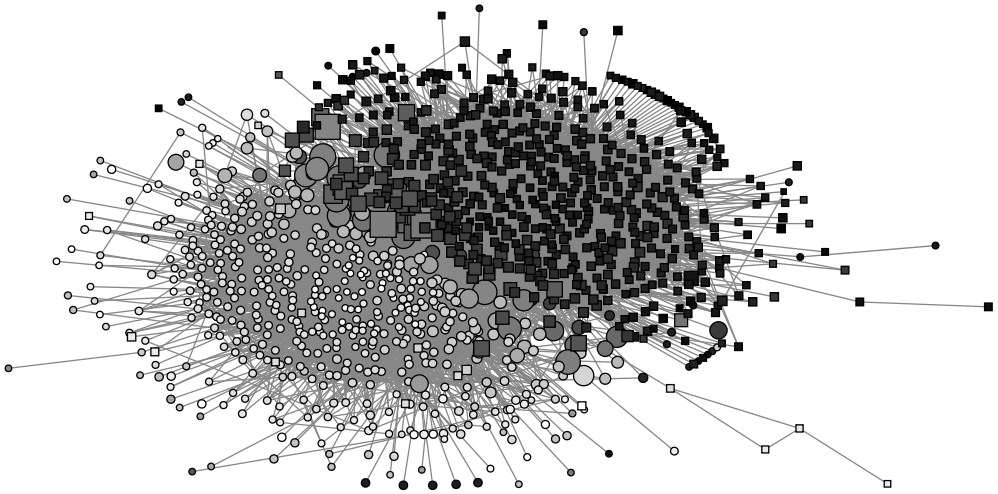


Figure 2: Fuzzy communities of the political blog network [7]. Circles denote liberal, squares denote conservative blogs. Dark shades represent vertices in community 1 (roughly representing conservative blogs), light shades represent community 2 (almost all liberal blogs). The size of the vertices is proportional to their bridgeness score.

pairs with weight less than 0.01 were excluded from the goal function. Therefore, out of the 746 031 possible vertex pairs, only 231 805 were considered.

Our fuzzy community detection algorithm was able to classify 94.84% of the weblogs correctly. 47 liberal weblogs were mistaken for being conservative and 16 conservative ones were grouped together with the liberal cluster. However, the mean bridgeness of the misclassified vertices was significantly higher (0.4575) than the correctly classified ones (0.2778). One must also take into account that the weblog labeling was admittedly imperfect (based on self-reported or manual categorization). The result is shown on Fig. 2.

IV. Conclusion

A possible extension of the fuzzy community detection algorithm of Nepusz *et al* [5] was presented. Our improvement eliminates a weakness of the original algorithm that assigns high bridgeness values to vertices lying on the periphery of the network, and by excluding vertex pairs with low weight from the analysis, the speed of the algorithm is also increased.

References

[1] M. Newman, A.-L. Barabási, and D. J. Watts, *The Structure and Dynamics of Networks*, Princeton University Press, 2006.

[2] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, 435(7043):814–818, 2005.

[3] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, 69:066133, 2004.

[4] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Publishing, New York, NY, USA, 1981.

[5] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó, "Fuzzy communities and the concept of bridgeness in complex networks," *Phys. Rev. E*, 77(1), 2008.

[6] M. Penrose, *Random Geometric Graphs*, Number 5 in Oxford Studies in Probability. Oxford University Press, 2003.

[7] L. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *LinkKDD '05: Proc. of the 3rd International Workshop on Link Discovery*, pp. 36–43, Chicago, Illinois, August 2005. ACM.

ANALYSIS OF AUTOMATIC TRANSMISSION CONTROL UNITS USING SELF LEARNING SYSTEM

Balázs SCHERER
Advisor: Gábor HORVÁTH

I. Introduction

In the automotive industry it is extremely important, to produce high quality, low failure rate electronic devices. Subsystem providers like Robert Bosch GmbH have to ensure the trustiness of their products for 5 to 10 years in a very harsh environment. The software development of Bosch automatic transmission control ECU-s (Electronic Control Units) is done in Budapest at the GS-TC/ENC-Bp department. To satisfy the above reliability goals this department employs more test engineers than software engineer. The goal of this PhD. work is to find a solution to improve the software quality of these TCUs (Transmission Control Unit) [1] and to achieve cost reduction in the test process by using a self learning test system.

II. Traditional test methods and phases

Traditionally there are two types of test processes at GS-TC/ENC-Bp, the so called regression test, which is done before every delivery to a car manufacturer, and the so called integration test performed after every modification of the software. Each of these test processes has the following main sub processes: I/O, CAN, Diag, Comm-diag, OMM, ROS.

The I/O test is responsible for testing the power supply and the analog and digital interfaces of the TCU. A general TCU has about 60 such I/O lines. Test measurements made on the power supplies of the sensor and actuator units, the proper working of the solenoids and ignition controls are also investigated.

The CAN test contains measurements for the CAN (Controller Area Network) [2] communication of the TCU. This test step checks the presence and periodicity of CAN messages (there are about 10 CAN messages containing more than 20 important signals), and whether the system is able to notice the absence of these messages.

Diag test deals with the so called filtering and de-filtering of different errors, this test checks whether the system is able to notice and store errors.

Comm-diag subprocess provides test steps for the diagnostic communication. In modern cars it is possible to read out some performance and malfunction related notes from the ECUs in a repair station using a simple diagnostic device. This step ensures that every diagnostic function (like KWP2000 [3] commands) works properly.

The OMM (Operating Mode Management) test step is responsible for checking the transition between the main operating states of the TCU, like initialization, drive, limp-home, shut-down etc.

The ROS (Realtime Operating System) tests the behavior of the RTOS by checking the stack usage, the periodicity of tasks and so on.

Each of the tests above is done in nearly the same environment. This environment contains a TCU, a so called Laborauto, which simulates the behavior of the car's other ECUs, and an ETAS INCA interface and software. The most important part of this set-up is the ETAS INCA [4] software and hardware interface. Briefly this tool provides a dual port RAM like interface to the TCU, and by using this interface every important internal variables of the TCU software can be examined on the

fly. Most of the above tests are done by checking, whether these internal variables contain the right values.

III. Goal and application space of a self learning based test system

As the previous chapter has shown there are extremely complex test processes at GS-TC/ENC-Bp, and many of these processes are under automation. However these tests are rather static ones. Currently there isn't any drive cycle based test analyzing the complete system`s behavior, simply because in this environment it is impossible to carry out such a test, due to the numerous variables and signals.

The goal of the self learning system is to fill this hole and provide a dynamic drive cycle based test.

IV. Difficulties of test system set-up

As the first step to create such a self learning system the base signal and variable set of the knowledgebase should be identified. As it was mentioned, traditional tests are done using INCA based monitoring of the states of internal variables, but unfortunately the set of these variables and their mapping are highly TCU version dependent. At GS-TC/ENC-Bp TCUs are developed for more than 5 car manufacturers, and there are several TCU versions for each manufacturer, therefore it is impossible to create a self learning system for such huge and changing knowledgebase. So as a decision INCA based diagnosis won't be used, and the TCU will be considered as a black box.

The next step was to separate a base of properties and variables that are identical to every TCU. The signals transmitted through the CAN communication seemed to be the best choice for this base variable set. However, the CAN matrix (the message IDs, and signal encoding) differs for each product line, but the set of signals are nearly the same for every TCU (like engine RPM and momentum, wheel speeds, accelerator pedal states, etc). This base set could be complemented with the diagnostic communications like KWP2000 protocol, or OBD [5] (On Board Diagnosis). These diagnosis protocols can complement the set of the variables provided by the CAN communication with information about the system states, and error codes, in a standard TCU independent way. Fortunately this black box configuration has very low hardware and software resource requirement and a test set-up can be realized in the Bosch Embedded System Laboratory at DMIS BUTE.

V. Conclusions, problems to solve

However, this set-up is promising and the knowledge base of the self learning systems can be collected, but there are two main problems to solve in the future. First of all there are no definitions for the correct or incorrect working of a TCU. Secondly there are many TCU versions, but even the same TCU version could have different behavior based on its so called calibration. And unfortunately this calibration can be changed even on the fly.

References

[1] The Bosch Yellow Jackets: *"Electronic Transmission Control ETC"* 2004.

[2] Robert Bosch GmbH: *"CAN Specification Version 2.0"* 1991.

[3] ISO 14230-2:1999. Road vehicles -- Diagnostic systems -- Keyword Protocol Part 1-4

[4] ETAS, INCA homepage: http://www.etas.com/en/products/inca_software_products.php

[5] On Board Diagnosis: OBD II, home page: http://www.obdii.com/