Model-based Containment of Process Fault Dissemination

Lambert Spaanenburg[1] and Martijn van Veelen[2]

[1]Lund University,
Dept. of Information Technology,
P.O. Box 118, 22100 Lund (Sweden)

[2] ASTRON,
P.O.Box 2
7990 AA Dwingeloo (The Netherlands)

Abstract *– The past decade has witnessed a marked increase in distributed system complexity. This was driven by a maturing technology that steadily decreased the number of faults. Unfortunately these fewer faults have become exponentially more costly. It becomes mandatory to detect faults prior to taking effect on the network. Such an early detection requires a new test detection and fault containment strategy, of which the outline and some basic ingredients are sketched here.*

Keywords *– Process Modelling; Fault Diagnosis; Abnormality Detection; Prediction; Neural Networks.*

I. Introduction

Modern society is foremost characterized by an increased sharing of information and communication in networks (or Grids). Of early date are the Transportation, Electricity, Water and Natural Gas Grids. Newer is the Information Grid (or Internet), quickly followed by information carrying networks for specific applications. This counts not only in support of the former named classical networks, but also in stimulating new sensory ones in Home and Industry [1].

Most of these networks grow without an overall architectural vision but rather by means of a local preferential attachment. This lack of predetermined structure does not mean that there is no structure at all. On the contrary, it has been noted that the seemingly chaotic self-organisation leads to a clear structure with distinct properties, though different from designed networks [2].

The default distribution of a programming error as part of the maintenance procedure, that in 1992 caused the New-Jersey blackout, may have seemed just that: an exception. But the problems kept coming back. The Allston-Keeler (July 1996) and the Galaxy-IV (May 1998) disasters gave rise to a concerted research activity on Self-Healing Networks [3]. In general the probable cause is a lack of investment on ensuring proper working conditions. A series of three disasters on the Electricity Grid in autumn 2003 (in respectively America, Sweden and Italy) shows that little progress has been made. And these are only the top of the iceberg [4].

Classical abnormality detection starts from the single fault assumption. Faults are detected at the output, but can hardly be distinguished from other faults on the same path to the system output. This makes the single fault a suitable representation for a chain of events, where one fault dominates the other. Such is not true in a network, where the information flow is not restricted to the forward path. Network faults command a new model, based on the cascaded consequences of the fault cause [5].

Such fault chains are characteristic for problem sources within and outside a network. They have become notorious for outside attacks, like in case of viruses. Hence protection against fault chains is often outward-bound leaving the interior largely unguarded. Hackers & viruses have known characteristics by which they can be identified, isolated and eliminated. Sometimes they are as easy as a suspect source address; sometimes the threat is coded deeper into the message [6]. In order for firewalls and similar measures to become also applicable to internal network causes, fault characteristics need to be determined before the cause starts spreading around.

The paper discusses the concept of and means for fault compartimentation. In section II the need for a separate detection view on a process is neurally motivated. Ensuing section III treats the notion of sensitivity, which is subsequently put in a neural perspective. This is illustrated by a limited experiment in section V and further concluded by a treatise on compartment. This sets the case for restraining fault dissemination in a network context.

II. Taking a proper view

The classical approach to Fault Diagnosis and Isolation (FDI) is based on the availability of a model for the working process. This model should be robust enough to provide reliable results despite noisy, irreproducible and incomplete measurement data. On the other hand, it should be sensitive enough to handle even unknown faults. Such a built-in conflict between model robustness and fault sensitivity cannot be completely resolved and must be balanced.

Inserting a single fault in the model and recording its effect brings a signature of the faulty process. Having a signature for the fault-free model and for models with a known fault (the *fault dictionary*) allows characterizing the process for the potential presence of such faults. Unknown faults cannot be handled, but should be off-line analysed and added to the fault dictionary for later usage. As the world of potential faults is unknown, the dictionary may grow in the course of time to uncomfortable dimensions.

The critique of the Self-Healing [3] approach is that a straight FDI turns the world upside down by directly classifying the fault instead of by detecting the abnormality first, followed later by an analysis of the nature of the cause. For a real-time network, the elimination of possible infections is needed immediately while the strategy for curing the network may come at a more appropriate time.

Faulty or abnormal behaviour can simply be defined as behaviour reaching outside the known good world. This implies that the model of known behaviour covers all aspects of the real world, which might not always be achievable, as the unknown world defies proper modelling by lack of data. Moreover, not every slight deviation of the modelled world needs to raise the alarm flag. With the usual noisy character of measurement data as caused for instance by the simultaneous switching of some large servers or machines, abnormal values can easily occur without having relevance to system faults. Simple methods are sufficient to clean up the network from the consequent alarm flood [7].

In this paper we assume that the measurement data are already cleaned from false indications. Only actual faults exist and will become noticeable as either change in the structure and/or in the complexity of the system function. We see these as the result of parameter variations over dimensions not included in the model. Such extra dimensions cover a number of related models, of which the fault models are discrete samples. This brings the need for two essentially antagonistic views on the same reality: (a) the *process model* reflecting the proper operation of the network, and (b) the *detection model* reflecting the occurrence of abnormalities.

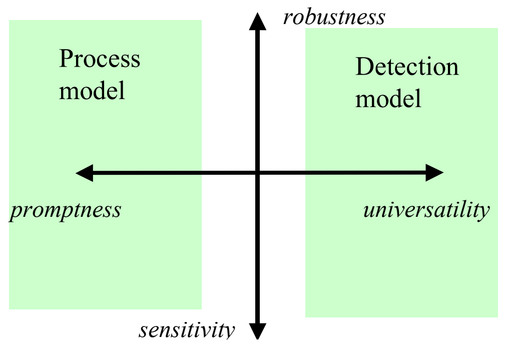

Figure 1. The parameter space for novelty detection and isolation (NDI).

The former relies on the extraction of physical coefficients from model parameters [8]. The aim is to find the Degrees of Freedom (DoF) in the model that describe the process in full complexity while minimizing the least square error (*risk minimization*). Domain experts usually add as constraint that the model can be humanly interpreted. The relevance of a known disturbance can be built-in by ensuring that the error on desired behaviour is a measure of the "faultiness" of the observed behaviour.

To handle unknown faults, the generic sensitivity (universality) needs to be improved by increasing the DoFs in the model. However increasing the DoFs conflicts with risk minimization. Simplifications may be called for to reach the desired separation of concerns. The justification for simplifications comes from properties on the independence of local processes allowing for modular models and linear behaviour. Indeed, if a system composed of multiple processes is in a stable equilibrium in its state space, linearization is allowed [9] and will break the dependence between sub-processes. Unfortunately these properties are insufficient as soon as the system drifts from the desired stable equilibrium.

The alternative proposed here is to break the dimensional dependence between model complexity and model risk to improve both overall sensitivity as well as confidence in the relevance of detected faults. Models used for detection must therefore extend beyond the desired process behaviour, intrinsically augmenting the model by extending the capabilities of the model beyond the DoF needs for describing the normal systems behaviour. The underlying philosophy is the multi-version technique, as popularized with great success in other technological fields. We argue that splitting the demands over two models seriously reduces (if not eliminates) the bias-variance problem that in the FDI era was an undividable part of abnormality detection. Such models need to match two opposing worlds by creating two different but dimensionally overlapping views on the same reality.

The question then remains what these dimensions are and how such effects can be brought to bear on the available model? To this purpose we introduce model capture by means of neural networks. A neural network extracts a model from presented data by means of a learning algorithm. By lack of an explicit model, learning will automatically establish the dimensions (or hidden features) that are needed to fit the process at hand and augment the mathematical model for detection purposes.

Any change in the structure and/or in the complexity of the system will affect the selection of the hidden features. In the following, we will elaborate on this theme. For the sake of clarity, we will refer in the following to the neural structure by its full name "neural network" instead of by the short hand "network" which we reserve for the information-processing commodity in general, being the process we need to monitor.

III. Sensitivity

For the purpose of our discussion we will use only fully connected feed-forward neural networks with a single hidden layer of neurons. The input neurons provide a direct mapping on the synaptic connections to the hidden layer. This layer reorganizes the problem space into a feature space that lies at the core of the neural model. Aiming at a minimal error for learning and recall, such hidden features may often not be physically plausible. Subsequently they are mapped on the synaptic connections to the output layer, which combines into the output result. These two synaptic connection layers (before and after the hidden neuron layer) have a clearly different meaning (Figure 2).

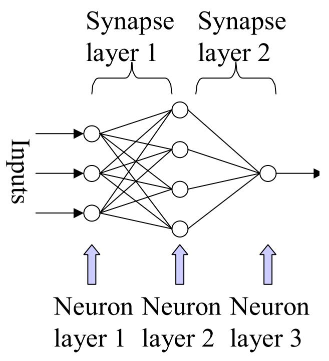

Figure 2. Layers in a neural network.

The hidden-to-output connections construct a solution with the hidden features as given; any modification that is possible on basis of the hidden features involves only a slight modification of the synaptic weights. Such could happen to accommodate innocuous variations in measurement data and takes a negligible amount of learning time to execute the small adaptation.

The input-to-hidden connections construct the basis for the neural model. When the basis is unusable for the neural model, a new one must be built by learning new values for the synaptic weights. This is a clear sign for the inadequacy of the model as captured from previous data and will consume an appreciable amount of learning time.

Such poses a crude indication for the presence of an abnormality by looking at the length of the learning period. An excessively high learning time clearly indicates the presence of a fundamental process change; a short time is merely adaptation (Figure 3). However, it appears that the two phases are not fully separable. Some of the hidden features are already being re-established while the others are still being accepted. In this phase, learning becomes slower but also unpredictable of duration [10].

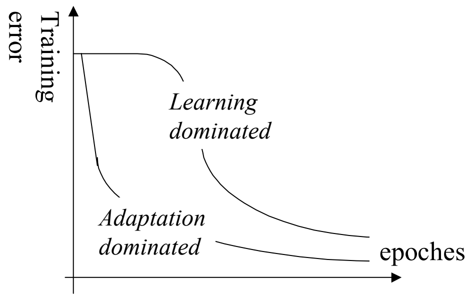

Figure 3. Adaptation vs. learning domination.

The hidden features do not identify the problem dimensions fully, but build a non-orthogonal base for combinatorial compositions to minimize the recall error. This provides the neural network with functional redundancy (*degrees of freedom*) as a number of different feature compositions can give the same overall functionality. By virtue of the non-orthogonal representation major input features can also be found as 2nd-order effects on various hidden ones. This observation can be extended to dimensions that do not appear in the mathematical model. If such effects appear in the presented learning examples, they will be learned. In other words, process aberrations, that are not primarily targeted for model capture, can still appear as higher-order effects on the hidden features.

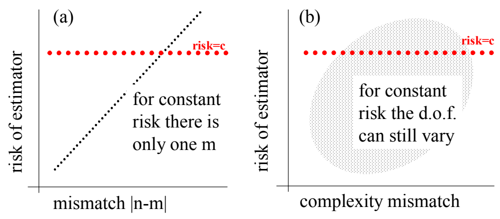

Figure 4. (a) regressive and (b) universal approximated modeling v.s. risk estimation.

The supporting reasoning on the (mis)match between the DoFs in the model and the complexity of the process is as follows. A process $y = a_0 + a_1 x + a_2 x^2 + \dots a_n x^n$ can be successively approximated with best fit parameters b as $y_{est} = b_0 + b_1 x + b_2 x^2 + \dots b_m x^m$. If m not equals n, we have a problem, as the model cannot be made more precise without affecting the risk (or error, see Figure 4a). When in contrast the process is universally approximated as $y_{est} = \varphi(\sum w_{kj}\, \varphi(\sum w_{ji} x_i))$ by for instance a neural network, the basis is formed by generic kernels, allowing the introduction of variables without changing the risk, provided the assignment can be solved. The inherent degrees of freedom provide a choice of target points (Figure 4b).

The first layer of synaptic weights determines the required selection of generic kernels. Model changes will therefore be directly reflected by such values. Hence it has been proposed to monitor model development by looking at the mapping on the hidden features. By nature of the learning method, such changes represent Brownian movements in the error space. This makes them unsuitable as a reliable and robust indicator. The alternative is the correlation between weights. On the network level, this method has received much acclaim to indicate that a signal variation is in line with others and therefore of chronic significance. Unfortunately the error landscape is very curvy; simple weight correlation will only characterize the scenery and not the itinerary. Additional attention on robustness is needed.

IV. Neural Robustness

The success of neural abnormality detection depends strongly on the ability to create a model for normality. On first sight, this may seem an impossible task as for several reasons the classification will never be accurate. Such is true but only in a numerical sense. Because of the non-linear curvature in the error space and the incomplete, irreproducible and noisy character of the learning data, a specific sample will almost never be 100% correctly reproduced. In a functional sense, this negative expectation is ill founded where clustering rather than approximation is being performed. Rather do we find here yet another occurrence of abstraction: a principle that rules universally in engineering.

Lets look, for example, at the clustering of analogue voltage values on the output of a logic gate. Though the "high" voltage levels will hardly ever be equal to "Supply" and the "low" voltage levels will hardly ever be equal to "Ground", the cluster based detection by a next gate will not see the difference between the many different values represented by "high" or between the many different values represented by "low". This learns that clustering is the basis for abstraction in the sense that any example represents the set.

The same principle can be applied in understanding neural networks. Each neuron builds a vector in the n-dimensional space, aiming to separate the data points into two parts with respect to one feature; a set of neurons separates the data points with respect to all features. Conversely, the data points are divided into clusters, separated by non-linear vectors. In order to understand this phenomenon, we take a short excursion into the physical prototype of force-directed clustering: the Earth/Moon gravity system ($F = g.m.M/r^2$ with g being some constant, m and M attributes of respectively the attracted and the attracting body (say mass, heat capacity or electrical charge), and r the distance between the bodies). Objects within the attractive force field of the Earth will only reach the Moon when the velocity is high enough, while objects that leave the Moon with a too low velocity will never reach Earth. Apparently, somewhere between the two bodies the total force will be zero: a meta-stable point in which the body will not move (see Figure 5a), but for any infinitesimally small displacement it will immediately leave in the direction of the field.

This is not actually what happens in every neural network. In self-organizing architectures, the winner-takes-all principle dictates an attraction mechanism that reflects the mass analogon, as discussed above; but in supervised feed-forward architectures the mechanism is more complicated, balancing both positive and negative influence. In other words, compromising between attraction and distraction. A typical example of such a mechanism is the model shown in Figure 5b.

This model originates from the physical behaviour of the micro-electronic diode. It assumes that the problem space is filled with many small particles of an either positive or negative attribute (say positive and negative examples). The intrinsic space is filled with the material in an evenly spread. This neutralizes the individual contributions within the overall effect. In fact, a natural tendency for global neutrality might be discerned where any disturbance will be dishevelled into the natural equilibrium. The effective field as shown in Figure 5b is what we would like to create by design in the neural network.

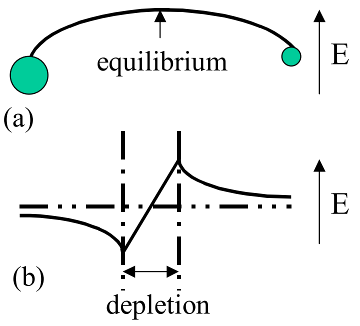

Figure 5. Attraction fields according to (a) the mass and (b) the diode analogon.

Where contradicting examples (positive and negative examples for the same context) are presented to a feed-forward neural network and no averaging compromise can be found, learning may momentarily halt or even definitively stopped [10]. Such error plateaus show by an almost Brownian movement to find an eventual escape.

It has been found that the detector quality is greatly influenced by redundancy appearing in many disguises. In coarse division one may distinguish between functional and structural ways. Most, if not all, of the functional redundancy scheme, read to a mass-oriented clustering, and will therefore not be interesting for this paper. For instance, a change in the presence of specific examples within the presentation set will only influence the field values but will not produce a diode-oriented scheme.

Structural redundancy is far more interesting for our purposes. It is based on having multiple neurons in the network structure for identical functionality. The impact of such structural redundancy has already been shown in experiments on reduction of the word width representing values in a hardware implementation [11], pointing out that by suitable measures the word width can be reduced to less than 8 bits. This gives further credibility to the observation that the detection robustness of the neural network can be enhanced by purposely introducing structural redundancy.

V. An experiment

VanVeelen has earlier introduced a technique [12] that combines the methods discussed at the close of section III by looking at the correlation between weight changes. The idea is that a persistent change needs to show up from the size and direction of the adaptations independent of the actual path taken. Apparently the first synaptic layer determines largely the detection quality of the derived model. Both its robustness and its sensitivity are relying on the judicious structural design and vice versa a careful examination of the weight changes may allow us to grasp the potential on-set of model changes.

An illustrative experiment is the one-step-ahead prediction of the sine function $v_n = \sin(n.2\pi f/T_s) + \mathrm{N}(0, 0.01)$ with $T_s = 32$ using 1024 data points with $f = 75/64$. The model contains 7 delay elements and 10 hidden neurons with linear output, 25 cross-validation models have been trained to stopping on the stable train error $\mathrm{C} \equiv |\mathrm{RSE}(\boldsymbol{w}_n, \xi_{\mathrm{train}}) - \mathrm{RSE}(\boldsymbol{w}_{n+1}, \xi_{\mathrm{train}})| < 10^{-4}$ for 10 subsequent epochs. The final RSE on the test sets ξ_{test} is $1.46 \cdot 10^{-2} \pm 3.2 \cdot 10^{-3}$.

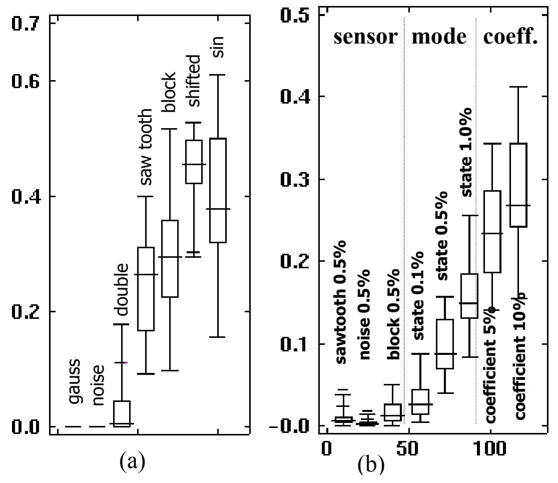

Figure 6. (a) Boundary gradient and (b) difference gradient correlation test for prediction of sine.

The disturbances are: (1) *gauss* - original sample; (2) *noise*; (3) *double* - $\mathrm{N}(0, 0.02)$ replaces $\mathrm{N}(0, 0.01)$; (4) *saw-tooth* - $0.02/T_s\,(n \bmod T_s)$-0.01 replaces $\mathrm{N}(0, 0.01)$ (5) *shift* $-f = 1.05 f$; (6) *sin* - $0.01 \sin(n.2\pi/T_s)$ replaces $\mathrm{N}(0, 0.01)$. We will apply the *boundary test*: fraction of gradients of possibly disturbed samples ξ outside the 2σ boundary, which is the standard deviation of gradients in response to a reference sample. The results are shown in Figure 6a. The additive chronic disturbances *saw-tooth* en *sin* are not significantly out of bound (around and below 5% as expected for random disturbances using 2σ boundary), while the internally changed process *shift* causes the gradients to go out of bound indicating a chronic disturbance. However the *double* disturbances would also suggest a chronic disturbance if this metric would be used, thus it lacks robustness.

The differences in a correlation test do not suffer from this robustness problem, as can be observed in the simulation results in Figure 6b. The largest observed fraction of different correlations in the 25 cross-validation models over all connections barely tops the minimal observed fractions of different gradient-correlations for the saw-tooth and does not even come close to the minimum fraction of different correlations for the shifted sine!

The question remains whether these disturbances are also observable by inspecting the recovery times. Unfortunately, in the shown example, the differences between the recovery times in all 25 cross-validation models are not larger than 1 epoch. As small chronic disturbances do change the model's gradient dependencies sufficiently to create havoc, this test has apparently missed the detection.

The robustness and sensitivity of gradient-dependency based detection relies on the presence of many related gradients. Figure 7 shows the distribution of the correlation coefficients for all connections to hidden neurons in an experiment on Volterra-Lotka models. Sufficient dependency seems to be present for detection.

The conclusion seems therefore justified that monitoring the recovery time of a learning system gives a quick glance, but needs for the present moment to be augmented by more elaborate scheme such as gradient analysis. This opens the door to in-line abnormality detection schemes, as discussed next.

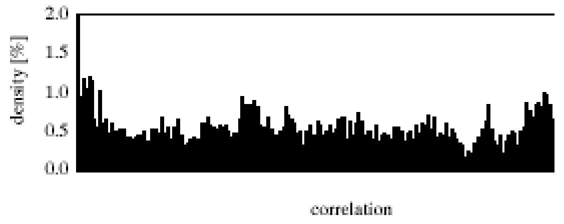

Figure 7. Distribution of correlations between gradients from connections to hidden neurons on a scale [0 1].

VI. Putting the picture together

In the previous sections we have discussed the need for a separation between process and detection model, and discussed some ingredients for an effective early detection of faults. We will now advocate the integration of such concepts in what we will call *containment*: an overall structuring that allows separation of concerns in all views, including on-line fault detection. In [13] it has been proposed that object oriented programming brings all the necessary ingredients. The classical control of actuator systems is based on hierarchical layers [14], where the higher layers can bridge defective units that are placed lower in the hierarchy. V.d.Klugt gives the example of Rudder Control, where functional redundancy is needed to safeguard the vessel and its passengers at all times.

This solves, however, only part of the problem. The early detection of process aberrations lies at the other extreme. In [15] it has been discussed that a system theoretical *attractor* must be constructed. Time-series of a single process parameter are cut into small overlapping series, called time-delay vectors (with a history of m tabs), and subsequently mapped into an m-dimensional space. This builds an attractor that will sizeably change shape if the process characteristics are slowly but fundamentally changing.

This provides the same sort of sensitivity, as discussed before. The learning process creates a sense of history, but taking care of a non-linear dependence and without the need to reduce the number of observed parameters. This is of interest, where the process interacts with its environment. In our case there will be different structures in different views. Another difference is the fact that the neural network evades the need to perform an m-dimensional mapping and matching. Instead we have a simple mechanism that can be easily added to an existing software object as a guard mechanism. We have to distinguish between two types of guards: off-line and in-line. The in-line guards are as discussed in [13] and are directly related to an existing software object. The off-line guards are required to handle faults that creep into the system from a non-functional part of the environment. This is depicted in Figure 8 as V-chart for the Power Grid control case discussed at length in [3] and are usually shaped as distributed agents. This picture has a striking resemblance to model separation shown in Figure 1.

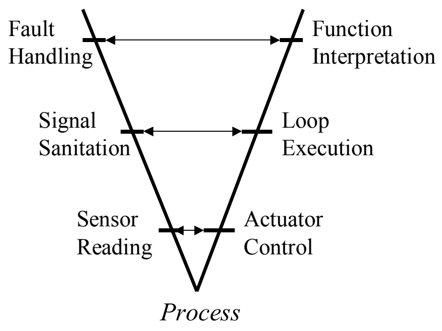

Figure 8. V-chart for separated process and detection modeling.

Most often, off-line network problems have a very basal cause. For instance, the quality of the electricity supply leaves nowadays much to be desired. Regularly, the supply is interrupted for a short time [16]. If this time remains below 10 ms, the human operator will not even notice it but the machines get affected at least to the degree of a noticeable reduction in lifetime. If the supply elapse takes longer, the electronic equipment gets deregulated. A much-advocated solution is the insertion of an UPS to bridge such gaps. This has caused a consequential problem in an Internet router, where the electronic equipment kept functional but the ventilator stopped, causing overheating of the system followed by a melt down. Future local supply systems can remedy this fault category, but still leaves other issues such as simultaneous switching untouched. Apparently, the network faults can become inserted at almost any time and place from other technologies than the mere electronic design. This makes multi-level modelling [17] a necessity to create a real robust operation.

We will use the name "compartment" for the integration of functional and non-functional objects over multiple views on the process in order to guarantee proper operation. As the compartment is loosely coupled to the functional hierarchy, there is a lot of choice in where exactly the compartment should be logically placed. In fact, the compartment is part of a detection hierarchy, logically linked to the functional hierarchy. It uses linked neural networks for early detection to integrate the respective views.

VII. Discussion

Abnormality detection was originally developed to monitor the control of industrial processes, but information networks have provided a natural extension. Though some niche successes have been claimed (for instance the detection of snooping attacks), the general breakthrough still has to come. The reason is on one hand the reliance on designed measures against outside attacks, and on the other hand the lack of distinguishable signatures for the unstructured internals.

In the meantime, network technology has broken through in industrial automation. Production lines with many sensors and actuators in distributed processes have emerged. This permits to extend the implementation whenever new technological possibilities and new insights in fault conditions become available. The historical development has often accumulated into a patchwork rather than into a physics-oriented blue print. Concepts like multi-level modelling [17] and automated configuration are much needed.

In a typical sensory network, a process part is measured and controlled. Sensors may be of a fixed, adaptive or configurable functionality. Despite such differences, the total effect is meant to provide predictable behaviour. It would be of advantage if such a functional behaviour can be handled without interference from other equipment. This both underlines the necessity to create compartments, but also forces to add a degree of autonomy to the concept.

For FDI purposes, the compartment will house both a formal process model and a data-driven detection model [18]. The detection model operates in conjunction with the configuration unit to identify the individual sensory components and their operational settings. This is partly because sensors may constantly be configured to different functions. Potential alarms are first filtered and subsequently diagnosed using the techniques described in this paper. Live alarms will then receive a classification by the mathematical model and accordingly be operated upon. Vice versa, the detection model may point to a need to update the mathematical mode for maintenance purposes. In this way the compartment provides linkage to the models we separated in the beginning of the paper.

VIII. Acknowledgement

Part of this work is inspired by the discussions on formulating the CAMERA project. Especially the discussion with Peter vanderKlugt (Imtech-Marine, The Netherlands) is sincerely appreciated.

IX. References

[1] Massoud Amin (2002) Modeling and Control of Complex Interactive Networks, *IEEE Control Systems Magazine* ***22***, Nr. 1, pp. 22-27.

[2] Albert-Laszlo Barabasi (2003) *Linked: How Everything Is Connected to Everything Else and What It Means*, Obidos.

[3] Massound Amin (2000) Towards Self-Healing Infrastructure Systems, *IEEE Computer* ***8***, Nr. 8, pp. 44-53.

[4] Massoud Amin (2003) North America's Electricity Infrastructure: are we ready for more perfect storms?, IEEE Security & Privacy, pp. 19-25.

[5] C.L. DeMarco (2001) A Phase Transition Model for Cascading Network Failure, *IEEE Control Systems Magazine* ***21***, Nr. 6, pp. 40-51.

[6] W. Bender et al. (1996) Technique for data hiding, *IBM Systems Journal* **35,** Nr. 3-4, pp. 313-336.

[7] Jonas Ahnlund, Tord Bergquist and Lambert Spaanenburg (2004) Rule-based reduction of alarm signals in industrial control, *Journal of Intelligent and Fuzzy Systems*, to appear.

[8] R. Isermann (1984) Process Fault Detection Based on Modeling and Estimation Methods - A Survey, *Automatica* ***20***, no. 4, pp. 387-404.

[9] G.J. Olsder (1994) *Mathematical Systems Theory*, Delfse Uitgevers Maatschappij b.v. (Delft).

[10] Emilia I. Barakova (1999) *Learning reliability: a study on indecisiveness in sample selection*, Ph.D. thesis, Rijksuniversiteit Groningen.

[11] Lambert Spaanenburg et al. (2001) Training neural nets for small word width, *Proceedings AmiRA'01* (Paderborn) pp. 171-180.

[12] Martijn vanVeelen, Jos Nijhuis, and Lambert Spaanenburg (2000) Process fault detection through quantitative analysis of learning in neural networks, *Proceedings ProRISC'00* (Veldhoven) pp. 557 - 565.

[13] Peter G.M. vanderKlugt (1997), Alarm Handling at an Integrated Bridge, *Proceedings IAIN.*

[14] R.A. Brooks (1986), A robust layered control system for a mobile robot, *IEEE Journal on Robotics and Automation.* ***2***, pp. 14-23.

[15] Ruud vanOmmen (2001) *Monitoring Fluidized Bed Hydrodynamics*, Ph.D. thesis (Delft University of Technology, The Netherlands)

[16] Edigna Menhard (2004) Schlechte Stromqualität birgt Sicherheitsrisiken, *VDI Nachrichten*, Nr. 9, pg. 25.

[17] Morten Lind (1994) Modelling Goals and Functions of Complex Industrial Plant, *Applied Artificial Intelligence* ***8***, no. 2.

[18] Jonas Ahnlund et al. (2003) Intelligent reduction of nuisance alarms in process control, *Proceedings ECCTD'03* (Krakow, Poland) pp. 369-372.