Decision Layer by Fusion of Diagnostic Algorithms

Jérôme Lacaille[1] and Tsirizo Rabenoro[1]

[1]*Snecma, 77550 Moissy-Cramayel, France*

jerome.lacaille@snecma.fr
tsirizo.rabenoro@snecma.fr

ABSTRACT

For manufacturers systems monitoring or production equipment optimization solutions are founded on specific algorithms that produce low level local information about risk of degradation or production loss. In either case local results are combined in synthetic reports aimed to help decision taking at higher level. This work is about the description of an automatic fusion mechanism able to build expert output with direct understanding of the system behavior and help to infer causes of efficiency loss. An example application was built and tested in a semiconductor fab. The algorithms diagnosed yield degradation in different subsystems or work-area and were digested in a weekly report that highlighted the main production problems. We deployed the same methodology for condition based maintenance of aircraft engines on a test platform. The first part of this document sketches out some notations, the second part describes the semiconductor application and the conclusion is dedicated to the transfer in the aeronautic domain for the decision level of an engine fleet health monitoring system.

1. METHODOLOGY

Our decision implementation is based on a Bayesian framework (Yu, Cleary, Osborn, & Rajiv, 2007) but is also known as a mixture of experts (Yuksel, Wilson, & Gader, 2012). A decision process is based on available information with confidence levels. Health monitoring information is mainly the result of computations of diagnosis algorithms. This result, for an algorithm a, is the probability that a score S_a reaches a given level θ_a. We call this probability a risk, R_a in Eq. (1):

$$R_a = P(S_a > \theta_a) \tag{1}$$

We often complete this risk value with precision information σ_a computed as an estimation of the output variance or a square error obtained by a supervised cross-validation process.

Each algorithm produces values from a variety of experiments. An experiment is materialized by a set of observations. For example one observes successive flights of the same aircraft and registers all parameters during the last month; this leads to a dataset $X_{aircraft}([t - 1\text{ month}, t])$. This dataset may be used by an algorithm that produces a risk $R_a(t)$. If we shift the observation window (one month) by one new day (about 10 flights), we obtain another lot of observations and the same algorithm will compute a new risk $R_a(t + 1\text{ day})$. Others results may also be obtained by modification of the algorithm's parameters. For example a change in the input sources, a change of a scale parameter, etc. We finally get a whole population of available results at a given time.

1.1. Fusion operator

Many results are linked together and from the analyst point of view it may be interesting to transform raw risk data into something more accurate that better corresponds to the underlying problem.

A fusion operator, Eq. (2), is a function that transforms a collection of risk results with precision into a new risk with a higher meaning for the analyst.

$$R_{a'} = \Phi_{a'}(R_{a_1}, R_{a_2}, \dots, R_{a_n}) \tag{2}$$

This seems to be another way to increase the risk population. But this process may be represented graphically and leads to better interpretation. We progressively replace low level risk computations by synthesis about subsystems or components.

If we use the case of successive diagnostics, then a fusion operator may also be some diagnostic confirmation that exploits the list of preceding computations to ensure the presence of degradation (e.g. the probability that one observes 8 detections over the last 10 flights).

1.2. Implication coefficients

Looking at the algorithm's parameters, its input sources and the observed dataset, we may associate each risk result to system component fault f with a weighting vector. The

implication value $\xi_a^f = P(F = f|S_a > \theta_a)$ may be interpreted as an a priori probability that fault f may occur when algorithm a result let us expect a risk of degradation.

The first implication vectors are defined by experts for initial algorithms which already are based on specific fault identification. It is more complex after some fusion operation. Moreover, as there may always be some dependencies between faults, even across different components, those implication vectors will be updated by a learning procedure using capitalization of experience.

1.3. Synthesis or projections

The decision process should use all available risk data. But the plurality of risk results makes it impossible. So we need to reduce the amount of information in a synthesis for each subsystem we want to monitor.

Suppose we have a component ζ that may be damaged according to a list of possible faults $f \in \zeta$. Let $\pi_f^\zeta = P(\zeta|f)$ the probability that fault f leads to a damage of component ζ. This value is often available from design and certification documents like the FMEA (Failure modes and Effect Analysis). A nice projection representing the risk of component ζ to be damaged may be given by Eq. (3):

$$R_\zeta = \sum_{f \in \zeta} \frac{\pi_f^\zeta}{\pi^\zeta} \sum_{a/\xi_a^f \neq 0} \frac{\xi_a^f}{\xi^f} R_a \tag{3}$$

This proposition is just a simple example using a logical weighted sum without taking care of faults and algorithms dependencies. Normalization constants π^ζ and ξ^f are used to ensure that the weighted sum has a value between 0 and 1 and keeps a meaning as kind of risk. Different fusion methods may also be used (like for example taking the highest detection). Eq. (4) below can be used to update the implication coefficients by a supervised learning procedure.

Independency of scores may be reached by designing algorithms related to subsystems. For example, in a fabrication process, if different but equivalent tools are used, local algorithms using only local measurements are independents. Using independent sets of faults and corresponding algorithms may lead to an exact formulation of a probability that a component is defective. Methodologies exist to group results into independent variables (Alhoniemi et al., 2007). Our actual goal is to converge to an approximate formulation with a Bayesian neural network. Other approaches may use dependency matrixes (Singh & Holland, 2010) or decision trees (Ricordeau & Lacaille, 2010) but at the end we always get a parametric model we may update with new experiments.

Fusion algorithm with risk result R_ζ is associated to the component ζ. If needed, we may compute implication values ξ_ζ^f from Bayes rule: $\xi_\zeta^f = \pi_f^\zeta P(f)/P(\zeta)$, the latter two probabilities are known and updated by maintenance people: $P(f)$ is the occurrence rate of fault f and $P(\zeta)$ is the repair or change rate of component ζ. The computation of this new implication coefficient serves the only purpose of being able to chain the analysis drawing a hierarchic graph of decisions from subsystems' components to higher subsystems and the global system that is monitored.

1.4. Confidence and efficiency

The main goal to achieve in decision making is gaining expert confidence. An optimization process manipulating the risk population is driven by this confidence. Gaining confidence is assured by a measure of the efficiency of the synthesis results.

Each expert may be viewed as a "human" or test algorithm e. So each expert produces one risk R_e usually linked to specific component's faults. Implication of the expert ξ_e is straightforward and given by the expert himself. However, an expert is subject to mistakes and so his precision σ_e is unknown and should be estimated. Usually, expert risk is a binary result: either faulty $R_e = 1$ or healthy $R_e = 0$.

In datamining processes, the expert result is known as supervision information. This supervision is used to build an efficiency measure like the mean-square error in Eq. (4).

$$\eta_a^2 = \sum_e \frac{(R_a - R_e)^2}{\sigma_a \sigma_e} \langle \xi_a . \xi_e \rangle \tag{4}$$

This value gives for any algorithm a measure of the adequacy between expert knowledge and the algorithm computation. It may be calculated for any component projection R_ζ and measure a confidence level for a decision about the health of this component.

2. SEMICONDUCTOR FAB YIELD LOSS ANALYSIS

An experiment at the ST-Microelectronics 8" fab in Rousset (France) showed that a great deal of information measured during the manufacturing process had a direct impact on the output yield. An influence analysis (measures of entropy and mutual information) measuring the stochastic dependences between the yield and automatically selected combinations of measurements proved the need to develop a prototype based on defectivity measurements.

A pilot project for collecting defectivity measurements and information on the routing of the wafers in the fab ran in real time at the Rousset fab (Lacaille & Dubus, 2005; Lacaille, 2008). It has highlighted many causes of wafer deterioration. Engineers and defectivity operators where able to automatically generate synthesis reports to supervise equipment health and production quality (Lacaille, 2005).

2.1. Microfabrication process

In a semiconductor fab, wafers are produced with a high rate (more than 1000 wafers a week). A single wafer is processed during almost three months and different products are manufactured at the same time by the same equipments. Wafers are grouped together in lots of 25, each lot is carried in a pod and all wafers in a pod are of the same kind. Sometime a lot is inspected, specific metrology equipments observe 4 to 6 wafers in this lot and we get defectivity measurements. Almost 20% of the lots are inspected for defectivity observations. This procedure allows only a very small proportion of the production to be observed. Moreover as defectivity operations are not productive steps, operators bypass them.

The defectivity metrology measures defects and produces observations like a number of defects of a minimal size, a defect density and a proportion of dies affected by the defects on the wafer area. Those defects are responsible of almost 80% of the yield loss in an industrial fab for a mature product.

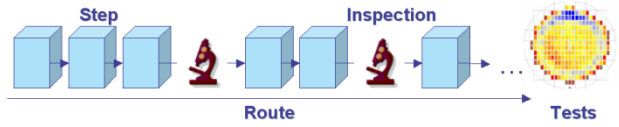

Figure 1 - Inspection steps are dispatched over the route of the wafers in the fab. The yield is computed at the end of the production by test steps.

In the 8" fab of ST in Rousset, defectivity operation steps are put on the wafer/product route but can be moved randomly to improve the metrology sampling on specific parts of the production (Figure 1). The route of the lot is determined by the product recipe but for each production step, each layer, the specialized operator can choose randomly between similar but certified equipments in his work-area.

To analyze the overall fab production in ST Rousset 8" it was necessary to model the random trajectories of wafer pods within the park of equipments. The stochastic nature of these trajectories is the result of the availability of similar and interchangeable equipments per work-area.

On those trajectories (simplified by successive arrows on Figure 2), defectivity steps appear like some measurement points (red dots) on a very complex curve. And route information, WIP (wafer in process) data, can provide observations on the same curve. For each step and each wafer we observe route information: date, equipment, recipe of the equipment, delay from the last step, slot position in the pod and sometime metrology information like defect density or defective dies. Route information is systematic while metrology is random.

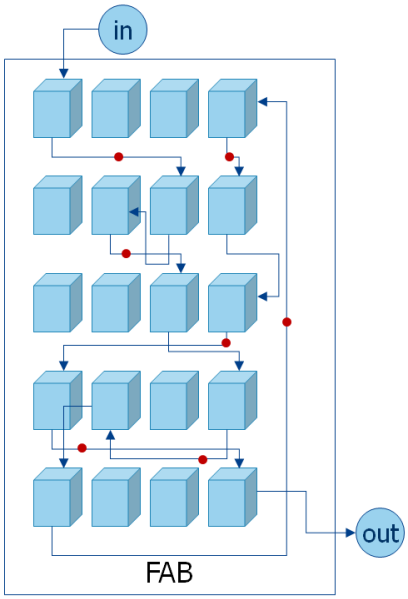

Figure 2 – Trajectories of wafers in the fab are complex curves (arrows) on which specific defectivity measurement points (red dots) are set.

The yield is measured at the end of the production in a specific back-end area by an electrical batch of tests. The yield information, results of those tests, almost corresponds to a proportion of surviving dies on the wafer.

2.2. Diagnosis methodology

The solution exploits a population of micro-algorithms: the agents. Those algorithms try to estimate a low-yield risk using some measurements taken on the wafer trajectories. This risk output R_a for each agent a is the probability of a low yield output.

Each agent uses only few points of measurements, thus ensuring the robustness of its calculation. In fact, the small analyzes carried out independently are not sufficient to explain the behavior of the fab, but a global information with a high degree of accuracy emerges from the entire population of local predictors.

Each agent uses some defectivity measurements taken during the route of the pods and/or process information taken from the tools (temperatures, pressures, etc.) and/or logistic data like equipment-id, receipt, inter-operation delays… The largest agent uses no more than 6 entries. The very large number of available measurements implies the existence of a lot of possible combinations of small number of measurements. The software solution solves this problem by using a population of agents distributed on a network of computers. This population evolves every day to adapt to the fab production.

This methodology is an implementation of machine ensemble theory (Figueiras-Vidal & Rokach, 2012) which

goal is to model complex functions using an ensemble of small data-driven components. One of the reference work in this domain is (Jacobs, Jordan, Nowlan, & Hinton, 1991).

2.2.1. Neural networks agents

The agents are self-adapting algorithmic components. Each one of them contains a batch of neural-networks regularly calibrated in real time. Each agent uses about fifty neural-networks (Figure 3). Each network inputs are the same inputs given to the agent; it is a small selection of WIP measurement points in the fab: defectivity, process values, and logistic data which are measured for each wafer. The neural-network models a relation between its inputs data and the wafer yield, the quality measure obtained by electric tests at the end of the fabrication process. Prediction error on calibration set is a quality indicator of the estimator. The wafer set used for calibration is a subset of the past processed wafers so the difference between two neural-networks comes from the "random" selection of the subset of wafers (see below). The final response of the agent is given by an estimation Y_a of the yield from a vote of a selection of the 20 best neural-networks. Then each agent a produces a low yield score $R_a = P(Y_a < \theta)$ and associated quality indicator σ_a as a final estimation of Y_a precision. The threshold θ is fixed according to the current fab expectation but may be adapted specifically for each product.

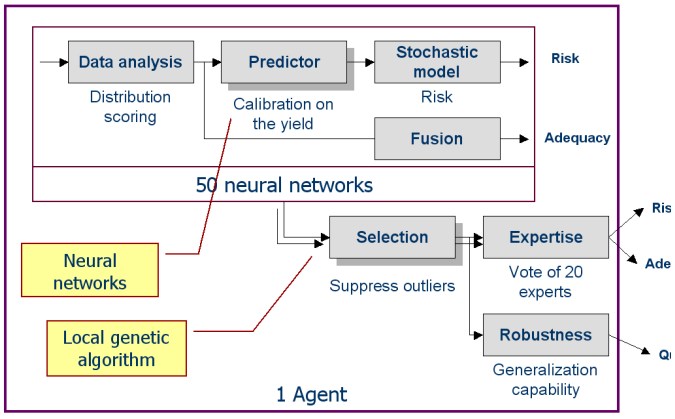

Figure 3 – Description of the inside of the agent. 50 neural networks, essentially multi-layer perceptrons with 3 layers, predict the yield from given parameters. Each neural network is learnt on a different training set ensuring accessibility to local behavior models. 20 among the best predictors are kept and used simultaneously in a voting process.

In this specific implementation the experts' skill is limited to the electric tests giving the real yield Y_0 and risk $R_e = P(Y_0 < \theta)$ for each processed wafer ($\sigma_e = 1$ or may be adapted by product). It is possible to compute an efficiency measure η_a from Eq. (4). This measure is used by the population optimization algorithm (Figure 4) to select the best agents.

The input observations sent to each neural network for calibration may be product-specific. Some agents work with all kind of products but the majority is specialized. A classification algorithm automatically selects the set of old products adapted for an agent based on the prediction efficiency on past measurements. Then new products are compared with preceding technologies using an unsupervised clusterisation algorithm to initiate implication values between agents and new products.

2.2.2. Agent population optimization

To make the agent population relevant at each time when the fab production evolves, we implement a genetic algorithm (Figure 4): the agents communicate together and exchange information on the relevance of each one of their inputs. The agent inputs are exchanged when agents meet together. The whole population is really moving over the fab and can be seen as a sort of swarm. The population of agent evolves gradually and it emerges an increasingly relevant response while leaving the system able to integrate each day some new characteristics of the fab production.

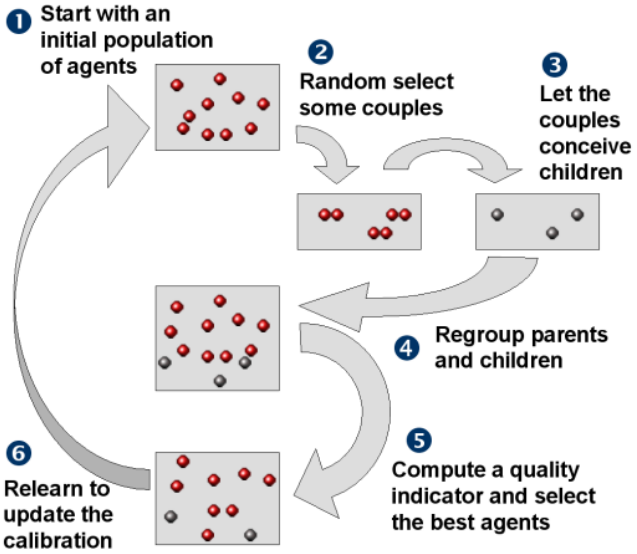

Figure 4 – A genetic algorithm optimizes the agent population.

The genetic algorithm optimizes the quality of the agent population but it has to respect hard constraints to ensure a homogeneous repartition of the agents over the whole fab and all products (see Figure 5).

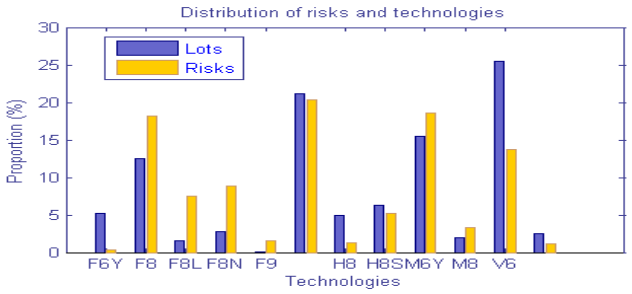

Figure 5 – A constraint forces the agents' population to be dispatched with a similar repartition to the production.

2.3. Decision help by fusion of diagnostics

A homogeneous population of agents emitting each one the same type of information replaces the complexity of the trajectories of the pods of wafers. The outputs of each algorithm are a risk of "low yield" R_a and a measurement of reliability η_a (Figure 6). Thus one can produce dashboards measuring the health of the equipments of work-areas, anticipating the quality of production and locating the equipments responsible for crisis and deterioration of the quality (Figure 8, Figure 9 and Figure 10).

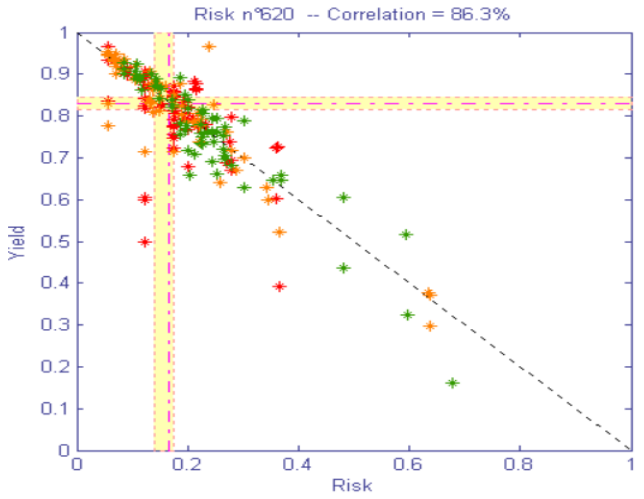

Figure 6 – Output of an agent. The risk (x-axis) is drawn with the yield (y-axis) for tested wafers. The green stars are computation of more than 66% of reliability, the orange have an reliability between 33% and 66% and the red ones are less than 33% of reliability.

Figure 7 schematizes the production of high level outputs from the original risk computations. The following figures are examples of such fusion algorithms.

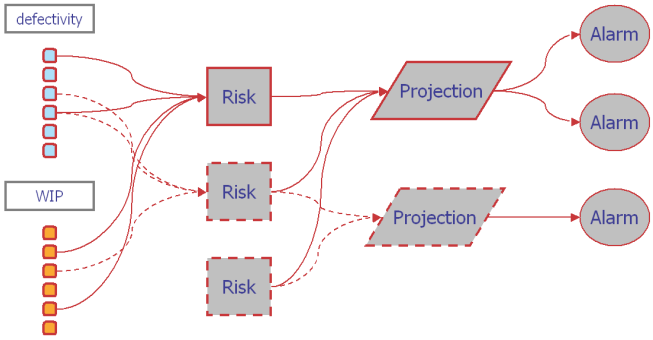

Figure 7 – Organization of the statistic computations. Risks are only intermediate results used by high-level fusion projectons.

Figure 8 is a fusion over an etch work-area. Here the fusion sums risks of low yield estimated for each wafer going through a given etch tool. For each tool, the summation is weighted according to Eq. (3) (with just one fault corresponding to the degradation of a die causing a decrease of the yield) and using implications that are high if the agent use information acquired on the given equipment, lower if not but if the wafer was processed by this equipment (depends on the distance on the wafer's route).

Figure 9 shows the evolution over time of the low yield estimation for a given tool using the same preceding computation. This graph helps to understand how to build alerts when a local risk crosses a maximum threshold or how to detect tendencies and eventually anticipate a crisis.

Figure 10 presents a synthesis for the worse equipments within all work-areas of the fab with recall of the past evolution.

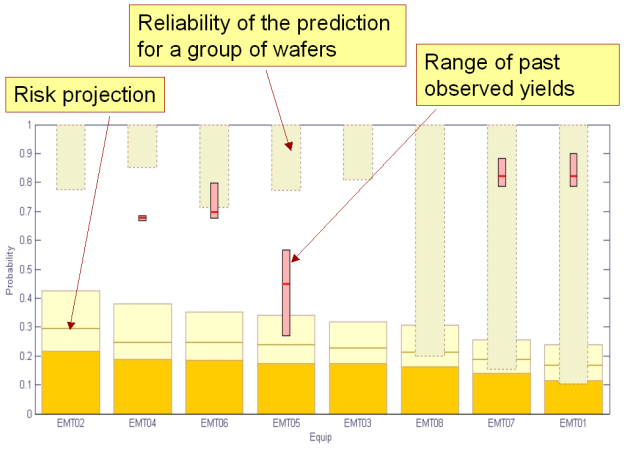

Figure 8 – High-level and detailed risk analysis of the equipements in a work-area. One have imediately a snapshot of the efficiency of each tool.

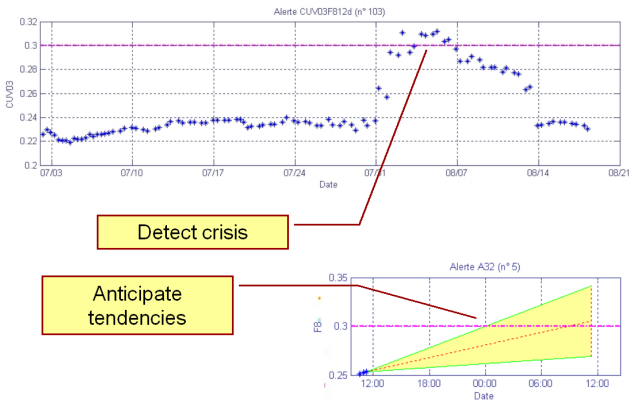

Figure 9 – Real-time survey of equipment. Alerts are automatically sent to operators.

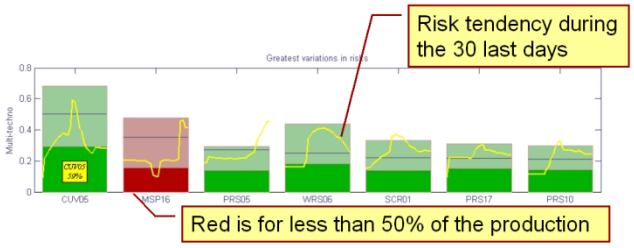

Figure 10 – Global analysis across all work-areas. All equipments of the fab are sorted by decreasing risks or decreasing risk-variations. The graph can also be filtered by work-areas or technologies.

2.4. Automatic inline scheduling

This whole system is entirely distributed over a cluster of computers (Figure 11). The agents are independent and their computations can be done in parallel. Only a central database for data collection (DC) is needed but it can be accessed simultaneously by different requests.

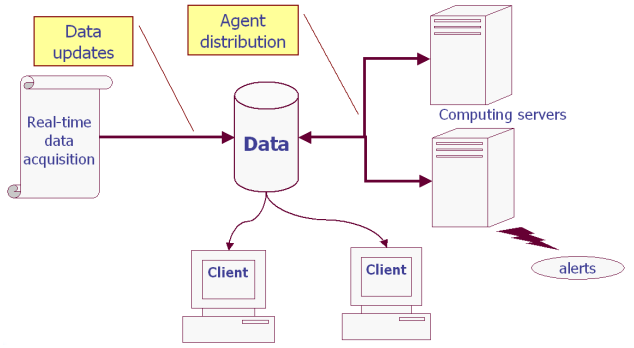

Figure 11 – Distributed architecture of the system.

The computation uses two internal cycles (Figure 12).

- The on-line scheduling cycle is based on the acquisition of new measurements. The new data are automatically pushed to the system by an ftp channel and the loader program automatically detects the arrival of new information.
- The genetic algorithm and the maturity of the agent population drive the internal scheduling cycle.

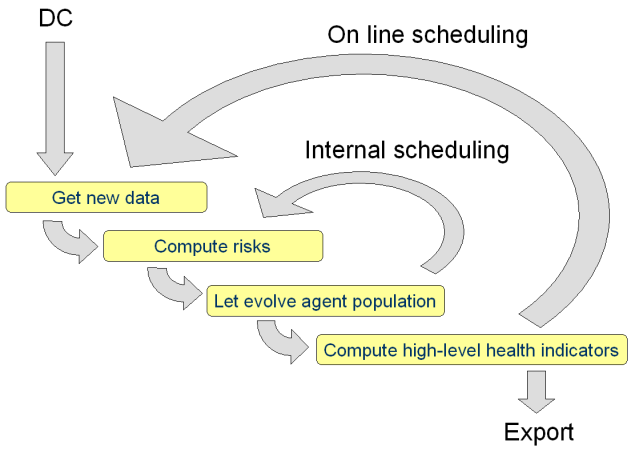

Figure 12 – System scheduling. The system is working non-stop ensuring a maturity of the agent population when the production evolves.

3. ENGINE CONDITION MAINTENANCE

Advanced health monitoring is becoming a standard for new engine applications, in order to enable in-service event reduction and engine maintenance optimization. The goal is to reduce operational events such as IFSD (In Flight Shut Down), ATO (Aborted Take-Off), D&C (Delay & Cancellation) and to substitute them with maintenance operations that are planned long enough in advance in order to minimize their operational impacts for the airlines. IFSDs and ATOs are very seldom but still stressing for the pilots and they often produce secondary damages that might increase reparation costs. D&Cs are usually not critical but occur more often; their consequences can be traffic disorganization, customer dissatisfaction. They are partially linked to procedures and controls to perform troubleshooting (Lacaille, 2012; Ricordeau & Lacaille, 2010).

The performance of engine health monitoring functionalities is driven by the capability to model the engine behavior and to identify engines on healthy or unhealthy conditions. Methodologies based on fusions of diagnostics applied to aerospace monitoring may be find in (Tang et al., 2009) for optimization of remaining useful life (RUL) estimations; in (Klein, Rudyk, & Masad, 2011) for the identification of bearing faults using vibration or acoustic signatures extracted from multiple microphones or accelerometers; and

in (Romessis, Kyriazis, & Mathioudakis, 2007) for performance analysis of the turbofan engine.

3.1. Health monitoring algorithms

Algorithms written for health monitoring purpose in Snecma are decomposed in two parts: the first one is made of embeddable code aimed to produce health indicators from engine raw measurements. Those computed indicators are sent to the ground with context information coming from the aircraft computer.

The ground application hosts the scoring process (Figure 13). The scoring process transforms health indicators into risk of abnormality R_a . This algorithm first uses a normalization phase to suppress the context dependency (flight specification like altitude, speed, weather conditions …) The second phase is either a model of normality which produces a likelihood to measure a range from normal behavior or an identification process using physical knowledge of the monitored system to target specific faults. At the end we have generic novelty detectors implying a subsystem in general or specific fault diagnostics (Lacaille & Nya Djiki, 2010; Lacaille, 2009, 2010).

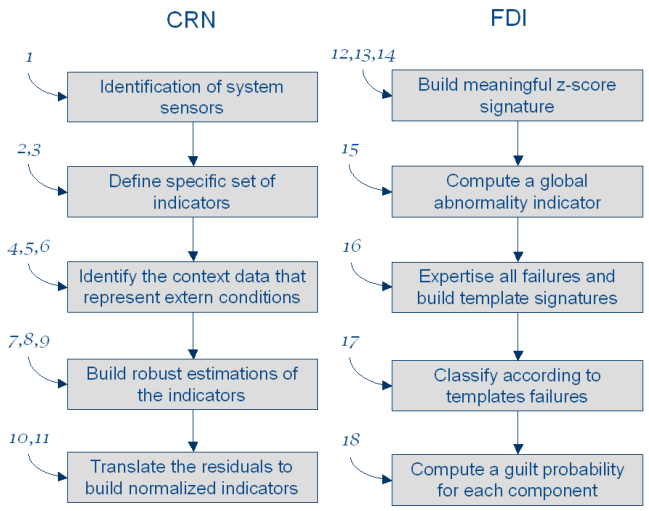

Figure 13 – Two phases of the ground diagnostic process. CRN (Context Removal and Normalization) suppress flight dependencies and FDI (Fault Detection and Identification) detects unusual behavior and try to identify the specific faults of a subsystem.

Precision σ_a of each algorithm is given by an estimation of the diagnostic error computed on a supervised dataset (Figure 14).

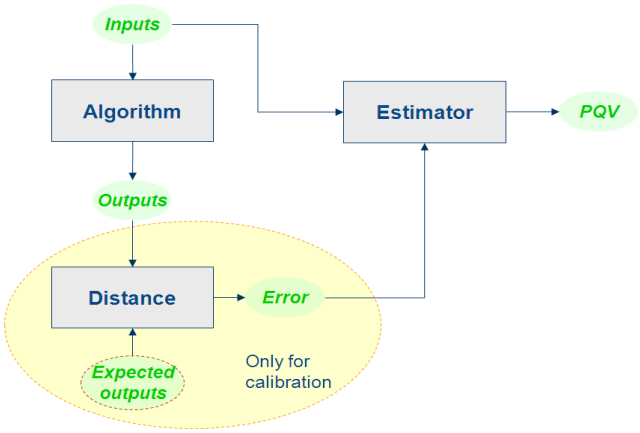

Figure 14 – Precision quality value (PQV) of the algorithms computed by an estimator learned on a supervision dataset.

The engine start capability (ESC) algorithm is a good application example (Flandrois, Lacaille, Masse, & Ausloos, 2009). Different subsystems and components are involved (Figure 15): the auxiliary power unit (ACU) that provides external pressure to the starter, the admission valve, the fuel pump, the fuel metering unit (FMU), the ignition system, etc.

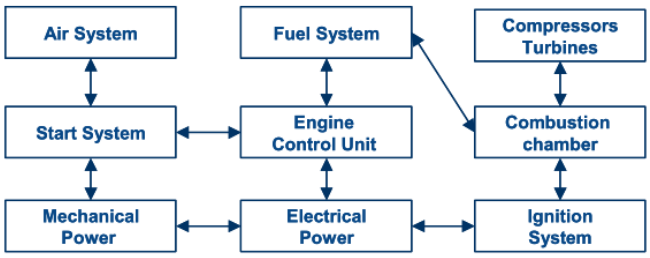

Figure 15 – Subsystems parts of the start system of a turbofan engine.

The health indicators are extracted from the raw temporal measurements with a “specific instant detection algorithm”. They are mostly time delays (Figure 16).

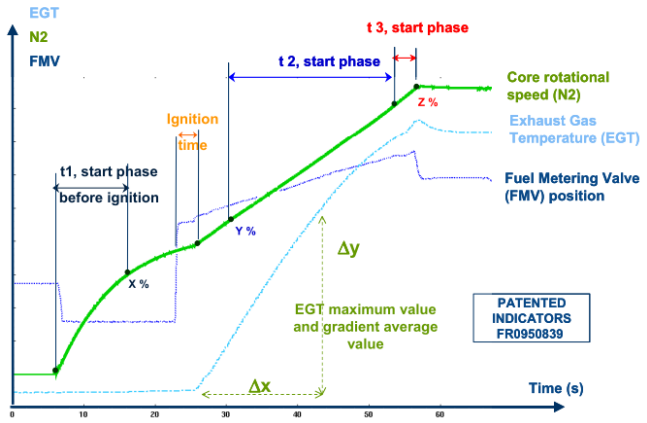

Figure 16 – Health indicators for the ESC algorithm.

Normality scores are computed for each indicator or coherent group of indicators if a multivariate observation is needed. At the end we build 8 scores listed in the following Table 1:

Table 1 – List of scores used for degradation risks computation of the ESC application.

Ref	Indicators
I1	APU Air Pressure
I2	Ignition time
I3	Exhaust Gas Temperature (EGT) gradient average value
I4	Exhaust Gas Temperature (EGT) maximum value
I5	t1 : start phase before ignition
I6	t2 : start phase between X % of N2 nominal value to cut out (SAV off)
I7	t3 : start phase to cut out to Y% of N2 nominal value
I8	N2 maximum acceleration (x,y)

3.2. Decision methodology

Decision step (on the ground) takes inputs from the set of risks generated by each specific algorithm. The specificity of algorithms comes from their design: there exist algorithms to detect each pump fault, gear wear, damaged bearings, oil leakage, filter clogging, intermittent sensors, etc. The faults relative to each subsystem produce comparable effects so the risk measurements are highly dependent.

The decision layer is actually built following the scheme presented in section 1 and derived specifically for a semiconductor fab in section 2. But in the case of aeronautic data we have a very small number of observed degradations, it is why the algorithms are mostly based on normality scores and their efficiency measured on a small number of observations. The implications come from expert knowledge but are progressively adapted using troubleshooting tools that records new data.

For the engine start capability (ESC) algorithm, the implication coefficients are roughly initialized by experts (Table 2). Decision on each impacted component is computed by fusion knowing the relation between faults and components.

Efficiency of the results is measured using maintenance reports on which identification of the problem is given after repair. Maintenance results are not always known by the manufacturer but as engine's design is completely mastered by the company; experts systematically give prognostics about the engines conditions. This supervision helps improving the implication values by a learning procedure.

Figure 17 shows an example of fusion of ESC detection algorithms. This fusion gives information about all components used in this process: APU, starter, valves, etc. This image presents the results of an artificial scenario where several defects were simulated sequentially.

The final decision is obtained only after some confirmation process that uses successive risk computations to solve incompatibilities and increases likelihood by a temporal confirmation.

Table 2 – Implication coefficients between faults and indicators. Values are roughly proposed by experts.

Failures	I'1	I'2	I'3	I'4	I'5	I'6	I'7	I'8
F1	0	0	0	0	0	NaN	NaN	0
F2	0	+1	+2	0	0	+1	+1	0
F3	+1	0	-1	0	0	0	-1	0
F4	0	0	0	-2	0	-1	-2	0
F5	0	+2	0	0	0	0	0	0
F6	0	0	0	0	NaN	NaN	NaN	0
F7	0	0	0	0	0	NaN	NaN	0
F8	-2	0	0	0	0	0	NaN	0
F9	0	-1	0	0	0	0	+2	0

+2	Very higher degradation as usual
+1	Higher degradation than usual
NaN	Not a Number
-1	Lower degradation than usual
-2	Very lower degradation than usual

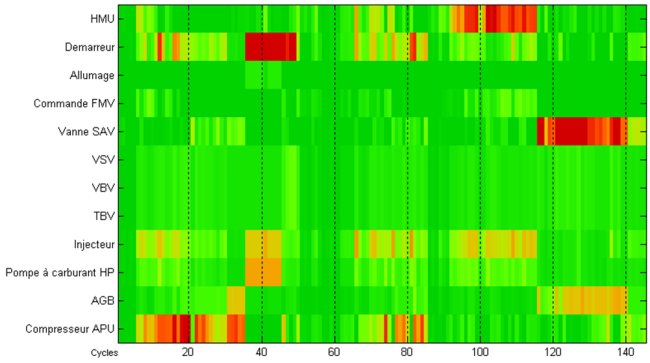

Figure 17 – Synthesis of the decision process for the different components that impacts the engine start system. Here the computed risk corresponds to the probability of a component failure leading to a D&C.

3.3. Conclusion

There is still a lot of freedom in the parameters used for normalization and identification algorithms as well as in size and shape of the confirmation by fusion of past computations. The methodology presented in the first section, applied in the second on a manufacturing process is compatible with engine health monitoring and is currently tested on Snecma's health monitoring research platform.

NOMENCLATURE

APU	Auxiliary Power unit
ATO	Aborted Take-Off
CRN	Context Removal and Normalization
DC	Data Collection
D&C	Delay and Cancellation
ESC	Engine Start Capability
FDI	Fault Detection and Identification
FMEA	Failure Mode and Effect Analysis
IFSD	In Flight Shut Down
PQV	Predictive Quality Value
WIP	Wafer In Process

REFERENCES

Alhoniemi, E., Honkela, A., Lagus, K., Seppä, J., Wagner, P., & Valpola, H. (2007). Compact modeling of data using independent variable group analysis. *Transactions on Neural Networks*.

Figueiras-Vidal, A. R., & Rokach, L. (2012). An Exploration of Research Directions in Machine Ensemble Theory and Applications. In *ESANN* (pp. 25–27). Bruges.

Flandrois, X., Lacaille, J., Masse, J.-R., & Ausloos, A. (2009). Expertise Transfer and Automatic Failure Classification for the Engine Start Capability System. In *AIAA InfoTech*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive Mixtures of Local Experts. *Neural Computation*, *3*(1), 79–87. doi:10.1162/neco.1991.3.1.79

Klein, R., Rudyk, E., & Masad, E. (2011). Decision and Fusion for Diagnostics of Mechanical Components. In *PHM* (pp. 1–9). Montreal (Canada): PHMSociety.

Lacaille, J. (2005). Mathematical Solution to Identify the Causes of Yield Deterioration. In *International Sematech Manufacturing Initiative (ISMI)*. Austin, TX: Sematech.

Lacaille, J. (2008). Global Predictive Monitoring System for a Manufacturing Facility. PDF Solutions. Patent US 2008082197 A1.

Lacaille, J. (2009). Standardized failure signature for a turbofan engine. In *IEEE Aerospace conference* (p. 11/0505). Big Sky (MT): IEEE Aerospace society. doi:10.1109/AERO.2009.4839670

Lacaille, J. (2010). Identification of Defects in an Aircraft Engine. Patent WO 2010076469 A1.

Lacaille, J. (2012). Validation Environment of Engine Health Monitoring Algorithms. In Snecma (Ed.), *IEEE Aerospace conference*. Big Sky (MT).

Lacaille, J., & Dubus, H. (2005). Defectivity Analysis by a Swarm of Intelligent Distributed Agents. In *AEC-APC*. Indian Wells (CA), Sematech.

Lacaille, J., & Nya Djiki, R. (2010). Detection of Anomalies in an Aircraft Engine. Patent WO 2010061136 A1.

Ricordeau, J., & Lacaille, J. (2010). Application of Random Forests to Engine health Monitoring. In *ICAS*.

Romessis, C., Kyriazis, A., & Mathioudakis, K. (2007). Fusion of Gas Turbines Diagnostic Inference - The Dempster-Schafer Approach. In *ASME & GT* (Vol. 1). Montreal.

Singh, S., & Holland, S. W. (2010). Trends in the Development of System-Level Fault Dependency Matrices. In *IEEE Aerospace conference*. IEEE Aerospace society.

Tang, L., Kacprzynski, G. J., Goebel, K., Vachtsevanos, G., Odel, B. A. M., & Echniques, S. E. T. (2009). Methodologies for Uncertainty Management in Prognostics. In *IEEE Aerospace conference*. IEEE Aerospace society.

Yu, L., Cleary, D., Osborn, M., & Rajiv, V. (2007). Information Fusion Strategy for Aircraft Engine Health Management. In *ASME & GT* (pp. 1–8). Montreal.

Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty Years of Mixture of Experts. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(8), 1177–1193. doi:10.1109/TNNLS.2012.2200299

BIOGRAPHY

Jérôme Lacaille is a Safran emeritus expert which mission for Snecma is to help in the development of mathematic algorithms used for the engine health monitoring. Jérôme has a PhD in Mathematics on "Neural Computation" and a HDR (habilitation à diriger des recherches) for "Algorithms Industrialization" from the Ecole Normale Supérieure (France). Jérôme has held several positions including scientific consultant and professor. He has also co-founded the Miriad Technologies Company, entered the semiconductor business taking in charge the direction of the Innovation Department for Si Automation (Montpellier - France) and PDF Solutions (San Jose - CA). He developed specific mathematic algorithms that where integrated in industrial processes. Over the course of his work, Jérôme has published several papers on integrating data analysis into industry infrastructure, including neural methodologies and stochastic modeling.

Tsirizo Rabenoro is currently a PhD student in statistics at Pantheon-Sorbonne University. He is also working at Snecma SAFRAN, as part of a research team involved in the operational monitoring. He is currently studying integration of mathematical methods into board and ground engine monitoring systems.