Software Requirements for Reliability-Centered Maintenance Application

Iony P. Siqueira
Companhia Hidro Elétrica do São Francisco, CHESF
Ensino Superior e Técnico em Informática, UNIBRATEC

Abstract—**This paper describes some software needs for Reliability-Centered Maintenance (RCM) application to asset management. Based on the RCM structured approach, a set of information artifacts and probabilistic data are derived as necessary for the correct application of the methodology. The method and experimental software have been used by a Cigré task-force in Brazil, as a supporting tool to develop a guide and database for application of RCM to power transformers. A concurrent research project supported by ANEEL (The Brazilian Electric Energy Regulating Agency) and CHESF (The San Francisco Hidro Electric Company) is also developing a standard format for exchange of information about RCM among power companies, equipment suppliers and independent system operators. Probabilistic optimization of maintenance frequency is a central requisite of these projects, as this aspect is loosely treated by RCM norms from IEC/ISO and SAE. A stochastic model is suggested and included as a software requirement.**

***Index Terms*—RCM, Reliability-Centered Maintenance, Software Requirements.**

I. INTRODUCTION

Among all contemporary technologies of asset maintenance, RCM (Reliability-Centered Maintenance) has expanded its application to practically all industrial sectors, achieving the status of preferred maintenance practice not only in aviation, but on nuclear and electric industries too. RCM is distinguished by its well structured process of analysis and decision, aiming the selection of maintenance activities. The method must be supported by a structured documentation process, for registration, knowledge management and auditing, in all its steps and used data. This requirement has motivated critics to RCM, due to the quantity of forms and related documentation, sometimes seen as bureaucracy. This aspect makes RCM an ideal candidate for software support, due to its unified approach and standardization.

It is the aim of this paper to present general requirements for an information system designed to support the implementation of RCM, according to international standards from IEC/ISO and SAE, and with documentation and organizational requisites of ISO 9000 standards. To adhere to current software technologies, the requirements should target a multi-user, client-server or multi-layer solution, to be used stand alone, or concurrently on local and wide area networks of large corporations. A beta version of its implementation is in current use by a task-force (SC-B3.01) supported by Cigré-Brazil, with contributions from many utilities, manufacturers, consultants, research centers and independent system operators, to develop an application guide of RCM to substation equipments. An ongoing research contract, financed by ANEEL (The Brazilian Electric Energy Regulating Agency), is supporting the development of information standards for inter-organizational asset management, covering RCM data requirements among users and suppliers of equipments.

The next part of the paper is a summary of RCM concepts and steps, related to the analysis of information requirements for a software solution. The third part, General Requirements, resumes the informational needs of Failure Mode and Effects Analysis (FMEA), as used by RCM, expressed as requisites in UML, the Unified Modeling Language standard of the OMG (Object Management Group). The fourth part, Activity Requirements, details software resources to support the decision process of RCM. Information Requirements are defined in the fifth part, by object oriented diagrams describing data structures used by RCM, and data database support needed. The sixth part, Modeling Requirements, details some stochastic models of failure modes used by RCM as software artifacts. The paper ends with some Optimization Requirements, in the seventh part, derived from the stochastic models, identifying data input and mathematical methods to achieve best maintenance results.

All records generated form a documental base sufficient for certification, knowledge management and auditing from maintenance engineering. Emphasis is given to support different statistical (multi criteria) decision models, for optimization and choice of maintenance frequency.

The paper concludes listing several types of reports needed by a typical user of RCM, such as FMEA, project control, maintenance plans, etc, in several formats, like SQL/XML/HTML, for intranet and internet publication, as well as standard office formats. Special requirements are related to interconnection to traditional CMMS (Computer Maintenance Management Systems) and inter-organizational data exchange for asset management.

II. RELIABILITY-CENTERED MAINTENANCE

Reliability-Centered Maintenance is a structured method to identify maintenance needs of physical and industrial processes [1,2]. Originated from aeronautical companies, in

I. P. Siqueira is with Companhia Hidro Elétrica do São Francisco (CHESF), Recife, PE, Brazil (e-mail: iony@chesf.gov.br).

1975, and supported by military industry in USA, RCM has been adopted by nuclear and electric industries, being applied in almost any modern industrial sector, nowadays. Besides recommending preventive activities, RCM also defines a consistent model relating each task to combat every failure mode. The approach involves answering a structured set of questions that identify the following treats:

- **Main functions** – what the user expects;
- **Functional failures** – losses of utility;
- **Failure modes** – failure causing events;
- **Failure effects** – failure dependent events;
- **Failure consequences** – resultant impacts.

Based on failure consequences (on environment, security, economy or process operation), the method suggests, through a structured logic, the most applicable and effective task to combat each failure mode, among the following options:

- **Time-Directed (TD)** – correct before failure;
- **Condition-Directed (CD)** – detect potential failures;
- **Failure-Finding (FF)** – uncover hidden failures;
- **Run-To-Failure (RTF)** – repair after failure.

Following RCM logic, the above order reflects a decreasing knowledge about the failure mechanism. The last task is recommended when the previous ones are not cost effective, and there is no security or environment issue; otherwise, a project change is mandatory.

III. GENERAL REQUIREMENTS

Any software to support an RCM process should include tools to document all its phases. It should supply not only a guided sequence to RCM logic, but support many operational (non-functional) requirements such as:

- **Integration** – with CMMS and maintenance software;
- **Storage** – in standard data bases management systems;
- **Importation** – from standard open formats (XML,etc.);
- **Exportation** – to standard formats (HTML,XML,RTF);
- **Standardization** – follow a recognized RCM standard;
- **Installation** – guided tour for RCM adoption;
- **Replication** – easy reuse of analysis results;
- **Documentation** – unlimited storage of each step data;
- **Multimedia** – inclusion of graphical and sound data;
- **Help** – on-line for each RCM step;
- **Auditing** – native tools for process auditing;
- **Security** – tools to access control and right permission;
- **Scalability** – from isolated to multi-user operation;
- **Management** – project follow-up of each item;
- **Optimization** – multi-objective task frequency decision;
- **Planning** – block aggregation of maintenance tasks;
- **Multiplicity** – analysis of several installations/systems;
- **Performance** – reduced use of network bandwidth;
- **Configuration** – of data-base and information sources;
- **Distribution** – of automatic version updates;
- **Centralization** – to easy network server consolidation;
- **Recovering** – from previous data base restoration;
- **Integrity** – by backup and compacting data tools.

This list is consistent with modern software requirements for corporate use.

IV. ACTIVITY REQUIREMENTS

Any RCM package should work in close relation to the organization Computer Maintenance Management System (CMMS), as shown on the UML Activity Diagram of Fig. 1. Note that the task of Maintenance Planning is designed as an RCM macro activity, dependent on the RCM Analysis of the installation project, and on information supplied by CMMS. Inside CMMS, this plan generates a Maintenance Schedule to be followed by execution crews. Failure Analysis occurs not only after each failure event, but also after each maintenance task. Unforeseen failures and statistics should trigger execution of RCM analysis, to review, validate or change current maintenance plans. Ideally, these two packages should function as a unique program. A multi-user system is recommended for large organizations, with geographically distributed installations, to maximize reuse of knowledge and expertise.

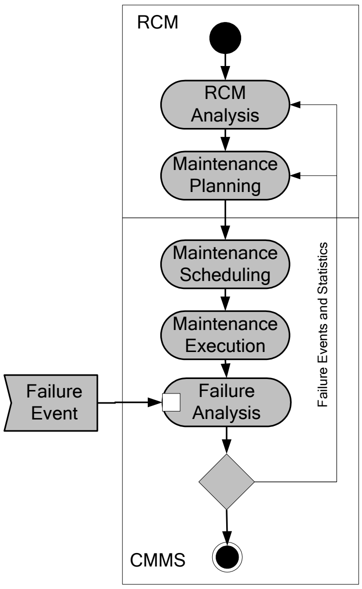


Fig. 1 - RCM and CMMS Interaction

Inside the RCM package, all steps are executed following a structured sequence, as illustrated on the next UML Activity Diagram (Fig. 2). Note that failure modes are initially identified based on physical components, while failures are initially identified after classification of system functions. This separation, previous to the FMEA step, allows their concurrent execution, possibly by different teams, attaining maximum speed to the process. The FMEA step reconciles these steps, relating failure modes and function failures to their effects and criticality.

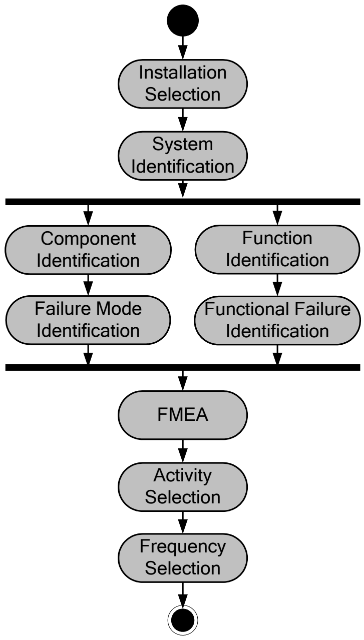


Fig. 2. RCM UML Activity Diagram

The Activity Selection step, following the FMEA block, obeys a structured logic, normalized by RCM, as shown on the UML Activity Diagram of Fig. 8. That logic conducts the analyst to decide on the most adequate and effective task to combat each failure mode. This includes traditional Time-Directed (TD), Condition-Directed (CD) and Failure-Finding (FF) tasks, as well as proposals for project change.

V. INFORMATION REQUIREMENTS

An RCM process uses much data and operating information about each project under analysis. Usually the following items are necessary as input, or to be generated from the target process:

- **Installations** – project identification;
- **Systems** – that compose each Installation;
- **Components** – that form each System;
- **Failure Modes** – of each Component;
- **Functions** – performed by each System;
- **Failures** – of each Function;
- **Causes** – for the occurrence of each Failure Mode;
- **Symptoms** – generated by each Failure Mode;
- **Actions** – possible to combat each Failure Mode;
- **Effects** – of each Failure Mode;
- **Consequences** – impacted by each Effect;
- **Activities** – selected to combat each Failure Mode;
- **Frequencies** – of execution of each Activity.

To support the necessary data management, a suitable static data model should be designed, such as the UML Class Diagram of Fig. 3. Observe the close relation among some classes and the corresponding activities in Fig. 2. This model allows the gathering of many supporting data such as component failure modes, failure symptoms and causes, general maintenance actions, as a reference database to be used by many similar projects. It also helps to structure libraries of previous RCM analysis.

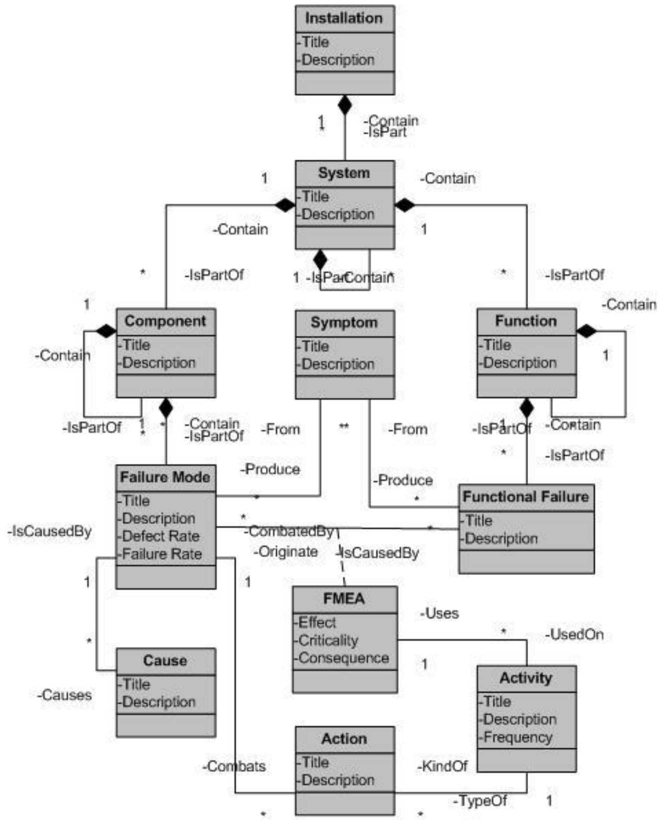


Fig. 3. RCM UML Class Diagram

VI. MODELING REQUIREMENTS

Application of RCM is dependent on the development of a suitable model for the dynamic or temporal behavior of each failure. To provide software support, models can be represented by UML state diagrams, from which similar Markov models can be derived.

The behavior of each Failure Mode is usually a stochastic process, governed by probabilistic events. Modeling of this process is usually a requirement for any RCM software. A general model must represent the dominant concepts of potential and functional failures. The first, as the detectable event of the start of a functional degradation, also known as a defect; the second as the inability of the item to perform its required function. These states can be viewed as slices of areas under the graph (Fig. 4) relating the item resistance to failure along its operating cycle, delimited by the following conditions:

- **Normal** – before a potential failure;
- **Defect** – between a potential and functional failure;
- **Failure** – after a functional failure.

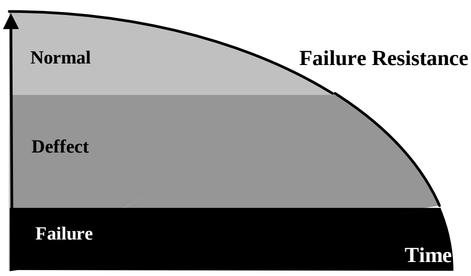


Fig. 4. Failure Resistance Curve

Besides these conditions, the item can be found on a state of preventive or corrective maintenance, according to RCM classification. Following this reasoning, five (most probable) states can be defined for an item, at any time:

1. **Normal** – apt to play its function;
2. **Prevention** – under preventive maintenance;
3. **Repair** – unavailable, under repair, after a failure;
4. **Potential Failure** – degradable state, after a defect;
5. **Correction** – under rectification after defect detection.

To derive a (UML) state diagram of this behavior, the following events are identified as causes for state changes:

1. **Prevent**: programmed inspection or maintenance;
2. **Correct**: action taken to correct a potential failure;
3. **Restore**: forced action to correct a functional failure;
4. **Defect**: partial or potential functional degradation;
5. **Failure**: forced functional interruption.

The term "**Correct**" refers to a planned event aiming to correct a known potential failure, before its evolution to a functional failure. It differs from a planned "**Prevent**" event where it is not known if there is a potential failure. The first three events are external maintenance events, while the last two are internal failure events.

These events trigger the transitions between the model states, as shown on the UML State Diagram (Fig. 5), where numbers from 1 to 5 are associated to each state.

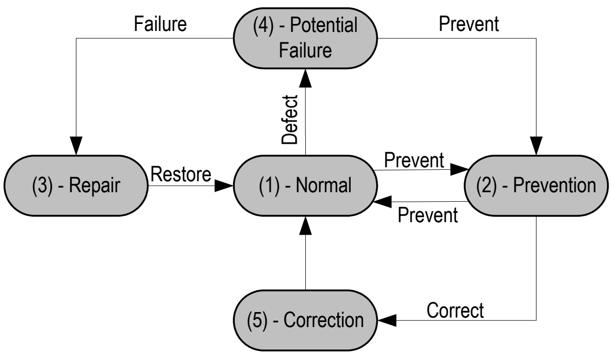


Fig. 5 - RCM UML State Diagram

A Markov model can easily be derived from this diagram, allowing the mathematical representation of the stochastic process, given their parameters. These can be resumed to two generalized parameters that define the failure mode behavior of each item:

- **Defect Rate** – probabilistic density of defects or potential failure in interval *dt*, conditioned to absence of defect at time t;
- **Failure Rate** – probabilistic density of functional failure in interval *dt*, conditioned to absence of failure at time t and presence of defect or potential failure at time 0.

Identification of these parameters in each population of items is a complex endeavor, in modern industrial systems, due to the progressive or hidden characteristic of most defects, with no evidence of the exact instant of occurrence. In consequence, failure rates ($\lambda_f=\lambda_{43}$), and defect rates ($\lambda_d=\lambda_{14}$), key parameters of the process, must be inferred from other observable variables, using the model. These variables are visible events and their duration, such as:

- Forced outage frequency ($F_f=F_{43}$);
- Preventive maintenance frequency (F_p);
- Corrective maintenance frequency ($F_c=1/T_{42}$);
- Mean time to maintain ($MTTM=T_2$);
- Mean time to repair ($MTTR=T_3$); and
- Mean time do correct ($MTTC=T_5$).

All of them must be retrieved (by estimation or statistics) from the connection of the RCM package to a CMMS system, as shown of Fig. 1. These are necessary data also for the optimization of maintenance frequency.

VII. OPTIMIZATION REQUIREMENTS

Any maintenance program based on RCM should propose a suitable frequency, for each preventive or predictive task. Data and calculations for optimal frequencies should be supported by any RCM software, based on well established theoretical models.

Maintenance optimization is achieved by determining values of task periodicity (T_{21}) and other parameters, such as Mean Times to Maintain (*MTTM* & *MTTC*) and Repair (*MTTR*), which maximize or minimize an objective function. Some variables must also obey some restrictions, such as physical viabilities, available resources and security requirements. As *MTTM*, *MTTC* and *MTTR* are limited by available technology, and assuming they are already at minimum values, the optimization must be sought by adjusting the maintenance frequency. The objective function must reflect the desired result, such as economy, risk, or quality of service.

The performance of any stochastic process may be estimated by a scalar indicator that expresses the cost/benefit of state transition in the system. Each transition *ij* can be pondered by a return coefficient (K_{ij}) that measures the gain/loss for the process for each maintenance event. That is:

$$I = K_p F_p + K_r F_r + K_c F_c \tag{1}$$

where I = scalar indicator or objective function;
K_p = preventive return rate (per event);
K_c = corrective return rate (per event);
K_r = repair return rate (per event);
F_p = preventive maintenance frequency;
F_c = corrective maintenance frequency;
F_r = repair frequency.

This is a general expression that can be applied to many indicators in industry [1]. Among them, the following are listed as examples, from power system practice:

- ***EFO*** Equipment forced outage;
- ***EFD*** Equipment forced duration;
- ***LPF*** Loss of production duration;
- ***LPP*** Loss of production probability;
- ***DNS*** Demand not supplied;
- ***PNS*** Production not supplied;
- ***EOF*** Equivalent outage frequency;
- ***EOD*** Equivalent outage duration;
- ***PDI*** Production discontinuity index;
- ***EVC*** Enterprise variable cost;
- ***CVC*** Client variable cost.

The return coefficients (K_p, K_c and K_r) measure the per-event cost or benefit contributed to the related indicator. Suitable statistics can easily be derived from historical data in a CMMS, to estimate these coefficients.

The ideal maintenance frequency that optimizes any of these indicators, taken as an objective function I, can be determined by expanding F_c and F_r on expression (1), as functions of the maintenance frequency F_p. These can be derived from the steady state Kolmogorov equations of a Markov model, replicating the structure of the UML state diagram of Fig. 5. This allows us to build a canonical non-linear programming system such as:

Minimize the objective function:

$$I = K_p F_p + K_r F_r + K_c F_c \qquad (2)$$

Subject to the restriction:

$$\frac{1}{MTTM} \geq F_p \geq 0 \qquad (3)$$

In these expressions, the repair (F_r) and correction (F_c) frequencies depend on defect (λ_d) and failure (λ_f) rates, which are functions inherent to process technologies and production environments. Values of *MTTM*, *MTTC* and *MTTR* depend on available maintenance technologies, and also affect these frequencies. The unique controllable parameter is the maintenance frequency. It can be null, if Run-To-Failure is the chosen RCM maintenance task, or greater then zero, in case of Condition-Directed, Failure-Finding or Time-Directed RCM tasks. The inferior and superior limits for F_p ($(1/MTTM) \leq F_p \geq 0$) are related to physical viability, as the maintenance frequency can not be negative or greater than the inverse of the mean time to do it. Figure 6 shows a typical plot for these expressions, as a function of maintenance frequency.

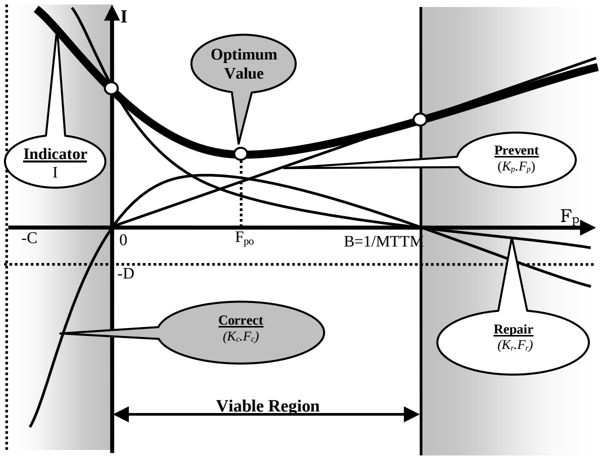


Fig. 6 - Quality Indicators

Note that the positive region of indicator I is formed by three parcels. The first, ***Prevent***, grows with the increase of preventive maintenance frequency, as a cost onus over the desired objective. The second parcel, ***Correct***, decreases with maintenance frequency, as a benefit brought by corrective maintenance. The third parcel, ***Repair***, ponder the effects of repair tasks on the indicator. This mix is typical of optimization problems, conducting to an equilibrium point among the parcels. The figure also shows how the three controlled parameters (*MTTR*, *MTTM* and F_p) affect the result. The greater the *MTTR*, the greater will be the ordinate of point *D*, and the frequency that minimizes the indicator. The same can be said of *MTTM* and point *B*.

According to classical methods to solve these systems, the optimum maintenance frequency will be given by a non-negative real root of the differential equation:

$$\frac{dI}{dF_p} = 0\,. \qquad (4)$$

By substitution, this equation reduces to a quadratic form, with two real roots [1,2,3,4]. Solving it gives the optimum maintenance frequency, for each failure mode.

Depending on equations parameters, both roots of equation 4 will be negative. In this case, the maintenance frequency that optimizes the objective function will be zero, at the border of the viability region. That is, only a Run-To-Failure strategy is recommended, as any preventive maintenance will degrade the objective function.

If, in addition to availability, there is interest in minimizing other indicators such as those listed before, a much more complex, multi-criteria decision problem (MCDM), will have to be solved. The same model will still be valid for each indicator, and a compromise solution will have to be negotiated among all system results.

As is usual with RCM activities, all results must be documented, for each failure mode and maintenance task. The optimum frequency will be a function of the following data, for each failure mode and task:

define the software needs.

After its testing in an extensible electrical transmission network, with more then 80 high voltage installations operated by CHESF, it is now being used by Cigré-Brazil Task-Force SC-B3.01, to define an RCM Guide for Substation equipments. A research project to standardize these requirements among several companies is under way, supported by ANEEL and CHESF, and conducted by CESAR, the Advanced Studies and Systems Center of Recife, Brazil.

Maintainability

- Mean time to maintain (*MTTM*)
- Mean time to repair (*MTTR*)
- Mean time to correct (*MTTC*)

Reliability

- Defect rate (λ_d)
- Failure rate (λ_f)

Productivity

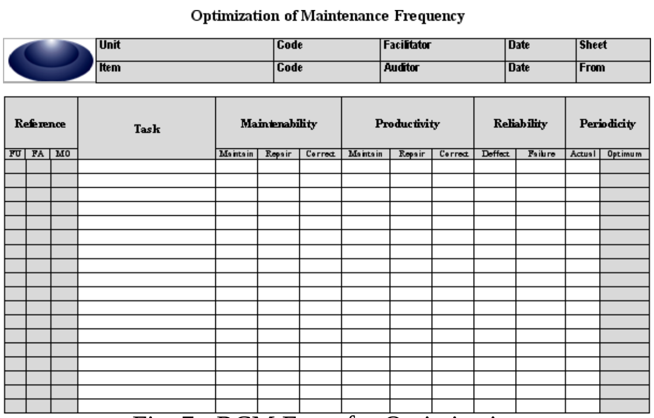
- Return rate to maintain (K_p)
- Return rate to repair (K_r)
- Return rate to correct (K_c)

Periodicity

- Actual maintenance frequency (F_p)

Excluding the last item, all other are difficult data to obtain, considering the lack of reliable statistical information about operating systems. For new items, good engineering estimation and judgment must be used to get initial data. Bayesian methods can be used to refine these estimations as experience is gained with the process. Accelerated life tests may be an option for small and inexpensive components. Once used for the first time, successive application of the model will improve the original estimation and results, as data from a CMMS system feeds back the RCM analysis step (Fig. 1).

The model can be formatted following the standard forms used by RCM, as shown on Fig. 7, or translated into a simple spreadsheet or an automated form as input to the RCM software to automate this calculus.



Optimization of Maintenance Frequency

Unit	Code	Facilitator	Date	Sheet
Item	Code	Auditor	Date	From

Reference			Task	Maintenability			Productivity			Reliability		Periodicity	
FU	FA	MO		Maintain	Repair	Correct	Maintain	Repair	Correct	Deffect	Failure	Actual	Optimum

Fig. 7 - RCM Form for Optimization

A research project financed by ANEEL is developing a standard data format, expressed as XML (eXtensible Markup Language) messages, to encode the information exchange among asset management systems of different organizations. These standard can also encode the data exchange between CMMS and RCM packages, as well as RCM data required from equipment suppliers.

VIII. CONCLUSION

This paper has shown general requirements for Reliability-Centered Maintenance software application. A statistical model of equipment defects and failures, and an approach to optimization of maintenance frequency is used as a way to define the software needs.

After its testing in an extensible electrical transmission network, with more then 80 high voltage installations operated by CHESF, it is now being used by Cigré-Brazil Task-Force SC-B3.01, to define an RCM Guide for Substation equipments. A research project to standardize these requirements among several companies is under way, supported by ANEEL and CHESF, and conducted by CESAR, the Advanced Studies and Systems Center of Recife, Brazil.

REFERENCES

[1] Siqueira, I.P., “Optimum Reliability-Centered Maintenance Task Frequencies for Power System Equipments”, 8th International Conference on Probability Methods Applied to Power System, Iowa, USA, 2004.

[2] Siqueira, I.P., “Manutenção Centrada na Confiabilidade – Manual de Implementação”, ISBN 85-7303-566-8, Editora QualityMark, Rio de Janeiro, Brazil, 2005.

[3] Siqueira, I.P., « Fiabilité dês Appareillages de Protection et son Impact sur lês Performances dês Réseaux », Conférence Internationale dês Grands Réseaux Electriques, CIGRÉ, Paris, 1996.

[4] Siqueira, I.P., “Processos de Decisão Markovianos em Sistemas de Proteção em Segurança”, MsC Thesis, Univsersidade Federal de Pernambuco, Brazil, 1999.

[5] Siqueira, I.P., “Impact of Protective Apparatus and Maintenance Scheduling on Power System Performance”, 4th International Conference on Probabilistic Methods Applied to Power Systems, Rio de Janeiro, ELETROBRAS, 1994.

[6] Lyonnet, P., « Optimisation d’une Politique de Maintenance », Technique & Documentation, Paris, 1993.

[7] Siqueira, I.P., “Análise de Riscos de Adiamento de Manutenção em Sistemas Elétricos de Potência”, XVI Seminário Nacional de Geração e Transmissão de Energia Elétrica, CIGRÉ, SP, 2001.

[8] Almeida, A.T., Souza, F.M.C., Org., “Gestão da Manutenção – Na Direção da Competitividade”, ISBN 85-7315-273-7, Instituto de Desenvolvimento da Engenharia de Produção – Recife, PE, 2001

[9] Moubray, J., “Reliability-Centered Maintenance”, Butterworth-Heinemann, London, 1994.

[10] Norris,J.R., “Markov Chains”, Cambridge University Press,Cambridge,1997.

[11] Aven, T., Jensen, U., “Stochastic Models in Reliability”, Springer, New York, 1999.

[12] Smith, A.M., Hincheliffe, G.R., “RCM – Gateway to World Class Maintenance”, Elsevier, New York, 2003.

[13] Bertling, L., "Reliability Centred Maintenance for Electric Power Distribution Systems", Doctoral thesis, KTH, Stockholm, Sweden, 2002.

[14] Hull, E., Jackson, K., Dick, J., "Requirements Engineering", Springer, London, UK, 2005.

[15] Booch, G., Rumbaugh, J., Jacobson, I., "The Unified Modeling Language User Guide", Pearson, New York,

USA, 2005.

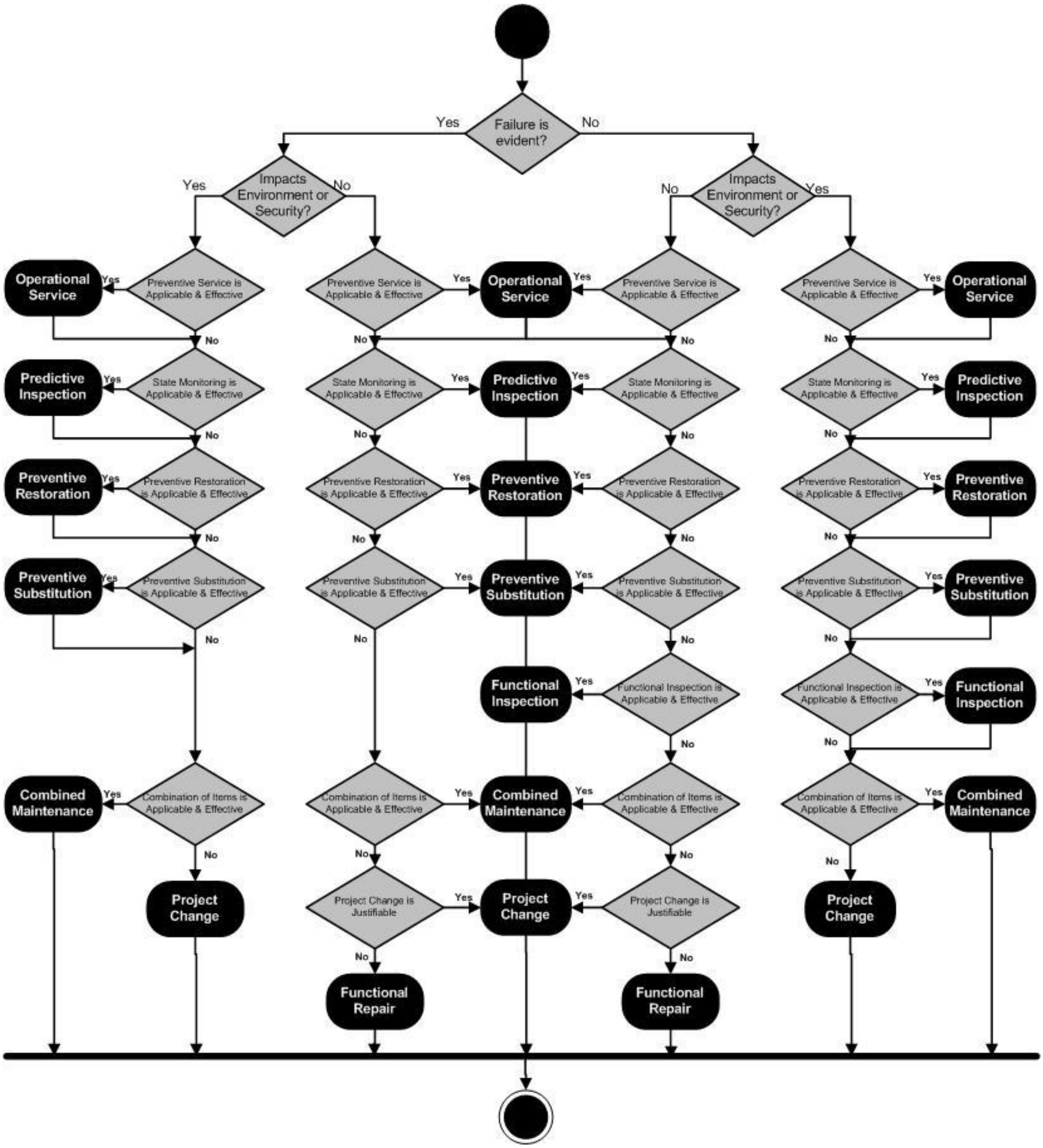


Fig. 8 - RCM Logic as UML Activity Diagram