Quality Problems

Automatic Intelligent Out-Of-Control Action Plan Proposal

The Goal is NIS extention with a new module for "Automatic Intelligent Out-of-control Action Plan Proposal".

It will be implemented in 3 sequenced steps, each step will be considered closed after field feedback; after each step closed it will be implemented the next. Following the 3 steps:

1. Quality problems Detection (fault detection issue)
2. Out-of-Control Action Plan (OCAP) definition (work-flow design issue)
3. Automatic Intelligent OCAP Proposal Algorithm (artificial intelligence issue)

Each step will be implemented using iterations; each iteration will have pre-fixed-time and will require a feedback from the user before begin next iteration.

Each iteration will give a part of the system installed and working.

1. Quality Problems Detection

The idea is to use Statistical Process Control methods based on control-chart. It will be used attributes control chart (p-chart, c-chart, u-chart) based on declared scrap items, or time between stops machine due to quality problems (using stop machines classification table available into NIS).

The system implements and monitors the control-charts and detects out-of-control of the same, following a simple flow:

1. system detects Out-of-control through control-chart
2. system generates an out-of-control message for the in-line user
3. the user confirms out-of-control, system save user feedback

Iterations Required

Iteration	Time Description (D=n.days)	Features
ITERATION 1.0	4D + 1D feedback-adjustment	1) evaluations of what kind of control chart to use 2) implementation of solution 3) implement GUI for fault signaling 4) installation on prototype machines 5) feedback (simulated OOC)
ITERATION 1.1	2D + 1D feedback-adjustment	1) based on feedback change/install 2) feedback (real OOC)

2. Out-of-Control Action Plan (OCAP) definition

Action Plan schema will be defined. **"Action Plan"** should be:

- Atomic Task (ie: lubricate gears)
- Sequence of Tasks (ie: increase pressure → reduce temperature → increase height)

NIS views each one (atomic task / sequence) as an "Action Plan". It will be possible to define Action Plan using Desktop Application from off-line side or using Devices Application (touch panels) from in-line side.

For this aim it will be implemented 2 Graphical User Interface (GUI) for defining Action Plan:

- the first usable off-line
- the second optimized for in-line usage

The same GUI will be used also for selecting existing Action Plan after a Quality Problem will be detected.

Iterations Required

Iteration	Time Description (D=n.days)	Features
ITERATION 2.0	2D + 1D feedback-adjustment	1) OCAP definition 2) implementation off-line GUI (inserting/selecting OCAP) 3) installation off-line GUI 4) feedback
ITERATION 2.1	4D + 1D feedback-adjustment	1) in-line GUI (inserting/selecting OCAP) definition 2) implementation in-line GUI 3) installation in-line GUI 4) feedback (simulated OOC)
ITERATION 2.2	2D + 1D feedback-adjustment	1) based on feedback refine GUI 2) installation 3) feedback (real OOC)

3. Automatic Intelligent OCAP Proposal Algorithm

The Algorithm will be able to suggest the better Action Plan based on "Quality Problems Detection Features" and "Previous Selection" by worker.

It will try to use 3 kind of algorithms:

- Rinforcement learning: based on the same artificial intelligence algorithm; NIS will be able to rinforce a Q-MATRIX "Features-Action Plan" based on worker feedback and a rinforcement function
- Bayesian Model: probabilistic approach, NIS will try to learn P(Action Plan|Features), P(Action Plan) using worker previous selection and will use Bayesian Model for probabilistic inference
- Logistic Classifier: gradient descent learing algorithm, using a logistic function for selecting an Action Plan considering "Quality Problems Detection Features"; learning of the model will be based on "previous worker selection" (it requires a lot of data)

In the first simple model: temporal sequence of Quality Problems will be considered "conditionally independent" (stationarity assumption).

Iterations Required

Iteration	Time Description (D=n.days)	Features
ITERATION 3.0	4D + 1D feedback-adjustment	1) features definition 2) implementation / test algoritms / selection algorithm 3) installation for off-line evaluation 4) feedback
ITERATION 3.1	2D + 1D feedback-adjustment	1) GUI implementation for Automatic Intelligent OCAP Proposal 2) GUI integration and installation on-line 3) feedback (simulated OOC)
ITERATION 3.2	4D + 1D feedback-adjustment	1) refining system based on feedback 2) installation 3) feedback (real OOC)

4. Prototype

Prototype Ready and some refinings will be required to adjust behaviour, usability, reliability of the system.

Finally:

- NIS detects Quality Problems
- NIS proposes ACTION PLAN
- Worker confirms Solving Problem ⇒ Positive Rinforcement
- Worker selects/adds Actions ⇒ Negative Rinforcement

Iterations Required

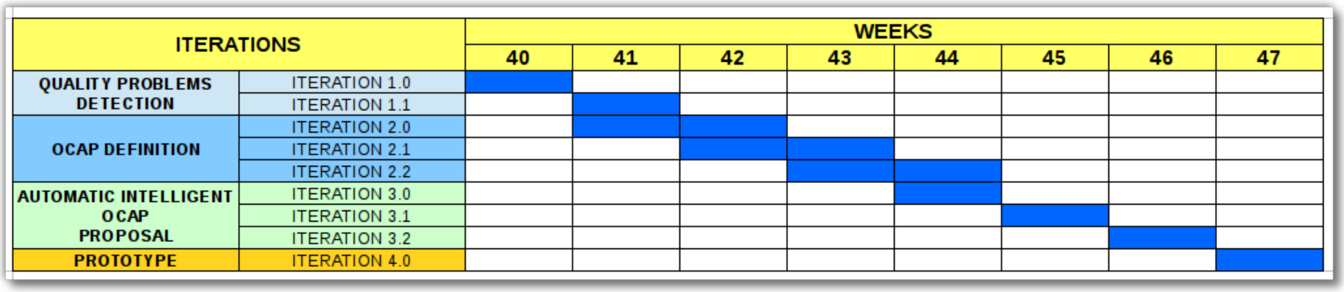
Iteration	Time Description (D=n.days)	Features
ITERATION 4.0	4D + 1D feedback-adjustment	1) refining 2) installation 3) final feedback

Table Of Time / Costs

It will be considered 2 workers for each iteration, so we have following Time/Costs table; man/day costs from Global Agreement Franke-NeXT.

Iterations	Extimated Time	Number of Workers	N.Man * Days	€ Man/Day	Costs €
1.0, 1.1	8	2	16	350,00	5.600,00
2.0, 2.1, 2.2	11	2	22	350,00	7.700,00
3.0, 3.1, 3.2	13	2	26	350,00	9.100,00
4.0	5	2	10	350,00	3.500,00
TOTAL COSTS					**25.900,00**

Expected Iterations Plan starting in week 40, it depends also by iterations feedback. Terminated each iteration releted features will be installed and working.



ITERATIONS		WEEKS							
		40	41	42	43	44	45	46	47
QUALITY PROBLEMS DETECTION	ITERATION 1.0	■							
	ITERATION 1.1		■						
OCAP DEFINITION	ITERATION 2.0		■	■					
	ITERATION 2.1			■	■				
	ITERATION 2.2				■	■			
AUTOMATIC INTELLIGENT OCAP PROPOSAL	ITERATION 3.0					■			
	ITERATION 3.1						■		
	ITERATION 3.2							■	
PROTOTYPE	ITERATION 4.0								■

Inizio Analisi Dati

Query Utili

```
SELECT MAX(m_rep_oee), MIN(m_rep_oee), avg(m_rep_oee), stddev(m_rep_oee),
MAX(m_rep_rf), MIN(m_rep_rf), avg(m_rep_rf), stddev(m_rep_rf),
MAX(m_rep_rv), MIN(m_rep_rv), avg(m_rep_rv), stddev(m_rep_rv),
MAX(m_rep_rq), MIN(m_rep_rq), avg(m_rep_rq), stddev(m_rep_rq)
FROM dimension INNER JOIN fact ON fact.codice=dimension.kfact WHERE
dimension.d1='FAFLA' AND D2='ORA' AND data>'20170101'
```

Pulizia database

```
TRUNCATE TABLE movimag;
TRUNCATE TABLE giacese;
TRUNCATE TABLE lottim;
TRUNCATE TABLE commass;
TRUNCATE TABLE dimension;
TRUNCATE TABLE fact;
TRUNCATE TABLE catego;
TRUNCATE TABLE assocate;
TRUNCATE TABLE catesup;
TRUNCATE TABLE statdef;
TRUNCATE TABLE statdefnc;
TRUNCATE TABLE ispdef;
TRUNCATE TABLE limits;
TRUNCATE TABLE formule;
TRUNCATE TABLE formvar;
TRUNCATE TABLE ubicaz;
TRUNCATE TABLE lotmap;
TRUNCATE TABLE caueff;
TRUNCATE TABLE caueffev;
TRUNCATE TABLE effect;
TRUNCATE TABLE ordprod;
TRUNCATE TABLE calendar;
```

Implementation

- Implementation