Ifeyinwa Obiora-Dimson, Hyacinth C. Inyiama, Christiana C. Okezie / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.1367-1376

Designing Multi-Agent Based Remote Process Monitoring Systems

Ifeyinwa Obiora-Dimson[1], Hyacinth C. Inyiama[2], Christiana C. Okezie[3]
[1,2,3] *Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka. Anambra State, Nigeria.*

Abstract

This paper presents an agent-based generic architecture for remote process monitoring. It is based on the fact that every process control system to be monitored can be represented in the form of an Algorithmic State machine (ASM) chart which is further translated into a corresponding State Transition Table (STT). By a little modification of the STT the entire process control system is represented as a succession of states each of which an agent can process. Thus by storing a copy of that table at the remote site and obtaining at each stage of the monitored process a feedback, this feedback information would serve as an index into the remotely stored table to crosscheck the validity of the feedback information with respect to each stage of the monitored process. Classes 0 through 3 state monitoring agents were defined depending on the number of qualifiers between the states they monitor and the next states. A process monitoring agent is also defined to activate the state monitoring agents when their services fall due and to further process their outputs as required by the stakeholders (manager, engineer, maintenance operator) in the process being monitored. There are as many state monitoring agents as there are states in the ASM chart all coordinated by one process monitoring agent. It is important to mention that this remote monitoring approach can be applied to any process even when the corresponding process control system is not agent based provided that a feedback is received at every state of the process being monitored. This agent based monitoring brings the advantage of object oriented analysis and design into the domain of industrial process monitoring systems development.

Keywords: Remote monitoring, process monitoring agents, state monitoring agents, real-time data capture, Graphical User Interface (GUI).

I. Introduction

The need for mobility and convenience often necessitates remote process monitoring and control. In this information age, people are getting used to using cell phones for various other applications other than making calls. Today, cell phones are used for anking transactions, information download from the internet, social network, environmental monitoring [1], and for many forms of group collaboration, from the convenience of one's location. The process control industries have begun to expect the same flexibility when it comes to process monitoring and control. For example, it is undesirable to force a maintenance technician or engineer to be glued to the VDU waiting for fault alerts. Today this is better done by sending the person a text message alert when one's attention is needed or making use of wearable personal computers [2]. In process monitoring, it should be possible to reach any stakeholder (technician, engineer, manager) wirelessly and supply all necessary information needed for appropriate decision making at one's level. Where a machine requires attention, this is often indicated by sending bytes of code from the process control wirelessly to any location convenient to the person. However, management information needs to be provided in an easy to digest manner. Continuous remote monitoring of the ongoing process followed by a Graphical User Interface (GUI) near the user that depicts the exact situation at the project site is desirable. It should also be possible to store captured data on the remote hard disk to permit playback or analysis of any segment of information defined by the time frame of interest [3]. The use of a Personal Computer (PC) or laptop for a GUI permits the presentation of the process information in a manner most convenient to managers. This should enable them make better management decisions.

This paper deals with real-time data capture, multi-agent based process monitoring, process management information and GUI.

II. Definition of state monitoring agents

Using object oriented software terminology; an agent is an object [4]. As an object it has a class it belongs to. In this paper, four object classes were recommended namely classes 0 through 3 [5]. A monitoring agent of class 0 makes a transition from its present state to another state without considering any qualifiers (fig 1a). Typically this happens where

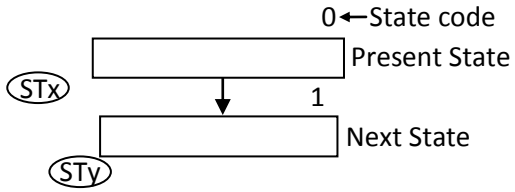

Fig 1a: state monitoring agent Class 0

two state boxes in an ASM chart are in sequence without any qualifier in between them. As shown in figure 1a, if the control system is in the state STX it must unconditionally transit to state STY when a clock pulse occurs.

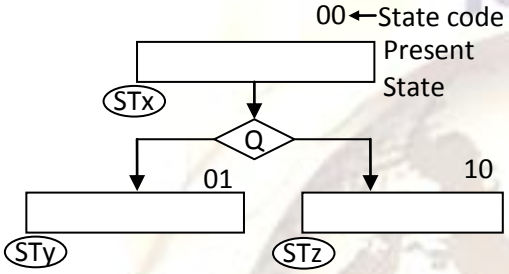

Fig 1b: agent Class 1

State monitoring agent class 1 monitors transition from one state STX to one of two alternative states (STY and STZ) depending on the value of the

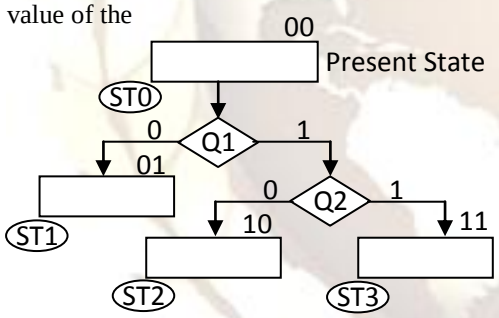

Fig 1c: state monitoring Agent class 2

qualifier (Q say). This is shown in figure 1b. If the process is in STX and the qualifier Q=0, the process transits to state STY. However, if qualifier Q=1, process transits to state STZ.

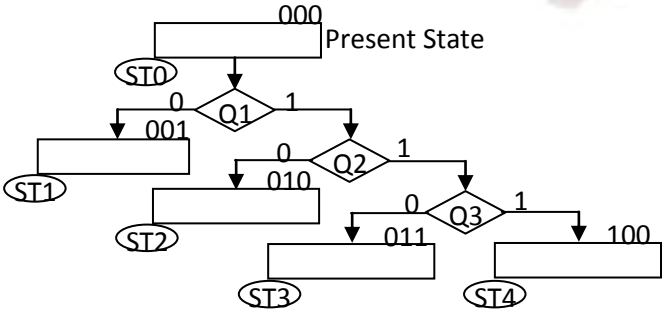

Fig 1d: state monitoring Agent Class 3

State monitoring agent class 2 has two qualifiers in cascade and the agent in the present state has four alternative link paths that determine the next state monitoring agent to handover to fig 1c. Each of the link paths may be selected depending on the values of the qualifiers Q1 or Q2 say. For example, if the present state is named ST0, the state monitoring

Table 1a: Class 0 monitoring agent transition

Present state	Next state
STx	STy

agent for ST0 would hand over to state ST1 if Q1=0. It would still hand over to the state monitoring agent for ST1 whether qualifier Q2 is 0 or 1. Therefore qualifier Q2 is said to be a don't-care. If however, Q1=1 then monitoring would be transferred to state monitoring agent ST2 if Q2=0 and to state monitoring agent ST3 if Q2=1 these transitions are shown in table 1a.

Table 1b: Class 1 monitoring agent transition

Present state is STx

Qualifier	Next state
0	STy
1	STz

Table 1c: Class 2 agent transition

Q1	Q2	State monitoring agent that takes over
0	0	ST1
0	1	ST1
1	0	ST2
1	1	ST3

Table 1d: Class 3 agent transition

Q1	Q2	Q3	Next state alternative
0	0	0	ST1
0	0	1	ST1
0	1	0	ST1
0	1	1	ST1
1	0	0	ST2
1	0	1	ST2
1	1	0	ST3
1	1	1	ST4

Note that a don't-care combination translates to two entries in table 1, the case when the don't-care is a 0 and when the don't-care is a 1 [6]. However both transitions are from state ST0 to state ST1 because of the don't-care status of Q2.

State monitoring agent class 3 is required for the condition where the present state is separated from the alternative states by up to 3 qualifiers in cascade (Q1, Q2, Q3) fig 1d. The transitions from the present state ST0 to each of the alternative states is determined by the three qualifiers Q1, Q2, Q3 as shown in table 1d. Note that when Q1 is a zero, transition must be to agent for state ST1 no matter what Q2 and Q3 are.

Similarly, when qualifier Q1=1 and Q2=0, transition must be to state ST2 irrespective of the value of ST3. The reason the fully expanded table is used is to facilitate the use of a look up table by the state monitoring agents which allows the indexing of the tables using qualifiers.

In practice, the next state alternatives are replaced by their state codes usually placed at the top right hand corner of the state box they represent rather than by their state names placed at the bottom left hand corner [7]. However the state names of the boxes are used in table 1 for the convenience of the reader.

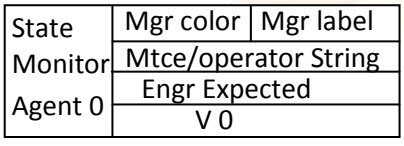

Fig. 2a: features of monitoring agent class 0

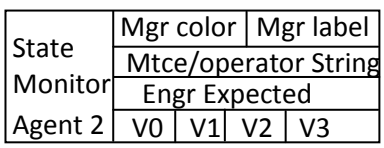

Fig. 2c: features of monitoring agent class 2

Although one could go on like this defining agent classes 4, 5, 6 and so on, the researcher did not consider this necessary because the optimization technique developed in this research allows on to limit the number of cascaded qualifier to at most 3.

2.1. Main features of a monitoring agent

Each state monitoring agent has the following features:

i. A color indicator (G, A, or R), G = Green, A= Amber, and R= Red, to be used when displaying for the manager to show him at a glance whether there is cause to worry or not. The color code (MgrColor) is immediately followed by a LABEL that describes what the system is doing at that state.

ii. A maintenance string or text used to give maintenance tip to the maintenance personnel or operator should a fault occur at that state.

iii. Hexadecimal digits showing the state code, qualifiers and the expected feedback byte for that state when the qualifiers are as shown. The expected feedback byte is compared with the actual when displaying diagnostics for the engineer at the bit-pattern level. Depending on the state monitoring agent type, the hexadecimal digits V*i* can be

a. 1 byte long for class 0 monitoring agents- no qualifier. (V0)
b. 2 bytes long for class 1 monitoring agents- 1 qualifier (q=0 or q=1) corresponding to (V0, V1) respectively.
c. 4 bytes long for class 2 monitoring agents – 2 qualifiers (q_1q_2= 00, 01, 10 or 11) corresponding to (V0, V1, V2, V3) respectively and
d. 8 bytes long for class 3 monitoring agents- 3 qualifiers (q_1 q_2 q_3= 000, 001, 010, 011, 100, 101, 110 or 111) corresponding to (V0, V1, V2, V3, V4, V5, V6, V7) respectively.

The features of the four classes of monitoring agents are features diagrammatically in fig. 2a through 2d.

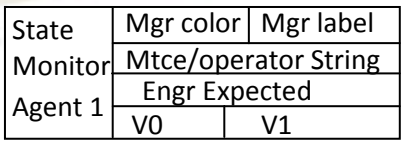

Fig. 2b: features of monitoring agent class 1

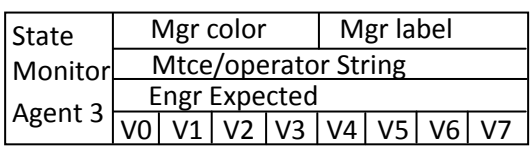

Fig. 2d: features of monitoring agent class 3

2.2. Process monitoring agents

By nature, agents are autonomous among other attributes [8]. Typically, the number of agents required to monitor a process represented as an ASM chart is equal to the number of states in that ASM chart. To allow that number of agents to have full autonomy may cause a loss of control in the system especially if something goes wrong with one or more of the agents [9][10]. Therefore the researcher has introduced the concept of process monitoring agents, where a process monitoring agent co-ordinates the activities of all the state monitoring agents in the same ASM chart. Recall that an ASM chart typically specifies a state machine in charge of a control process.

Thus, a process monitoring agent obtains the inputs from the process being monitored via feedback, extracts the present state code, activates the corresponding state monitoring agent and supplies it with the values of the qualifiers in use. The activated state monitoring agent would then produce an expected output pattern to be compared with the actual feedback received. The state monitoring agent could also release the feedback information contained in the manager string or maintenance string. The information supplied by a state monitoring agent can either apply to the engineer (engr), the manager (mgr) or the maintenance personnel (mtce) depending on the setting of three radio buttons as shown in figure 3.

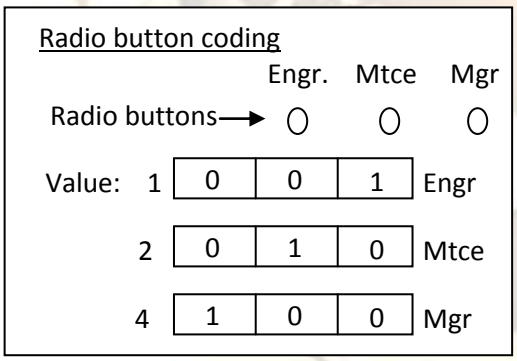

Fig. 3: radio buttons for selecting user specific information

When the activated state monitoring agent finishes its assignment, it deactivates itself or sleeps until it is activated again by the process monitoring agent. No state monitoring agent that is not activated by the process monitoring agent would function. The process monitoring agent uses this fact to activate the state monitoring agents one at a time depending on the feedback information received from the control system.

III. The Graphical User Interface (GUI)

This is a visual aid to the operator, manager or engineer to enable them understand at the remote end how the process under control is progressing dynamically. Certain forms of GUI that depict the process progress in a way that the end users are familiar with are preferred. These include:

a. State Transition Table (STT) based analysis of process progress.
b. Bar charts that rise and fall in line with process progress with a unique colour for each state in the process.
c. The ASM chart of the process under monitor which is animated with the state code and qualifier bit pattern. In this case, the colour of the active state box changes from state to state as the process progresses. Similarly, the active qualifiers and the link path their logic values recommend are highlighted in colours as the process progresses.
d. It is also possible to have a combination of the bar chart followed on the side by appropriate decision boxes and their link paths.

It is important to note that where the process cycle is too fast for the human eye to follow, the monitoring software may deliberately lag behind the process if desired. In this case, all the necessary data capture is made but the play back in the GUI is made to follow the pace the human eye can track for the benefit of the users. This is especially useful when carefully analyzing relevant segments of the data capture in order to locate anomaly.

A radio button is provided on the screen of the GUI permitting a user to select one of three possible display types, namely:

a. Display for manager: contains colour bar and its label showing what happens at that state. Three colours are used: green, amber and red. Green implies normal operation, amber means that automated test is being done that could fail, red implies a more serious condition, perhaps waiting for an operator input that has failed to come so far.
b. Display for operator or maintenance personnel: a maintenance error correction tip which if not successful would require the attention of the engineer.
c. Display for engineer uses bit pattern level to depict process output that did not match expectation.

IV. Steps in SMS to database software subsystem

The steps in SMS to Database software subsystem include:

1. Feedback interrupt (FB-INT) from SMS interface
2. Get feedback from SMS interface
3. Append date and time
4. Store in the database for feedback

5. Loop back to 1 and continue forever.

The block diagram representation of the steps is as shown in figure 4.

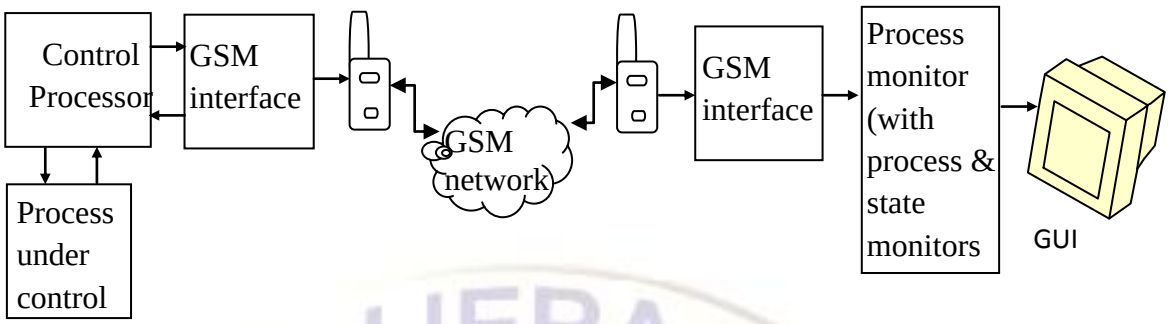

Fig 4: Block diagram for SMS to database software subsystem

Fig 5 shows the flowchart for feedback data capture from the process under control. Each feedback record has date and time of arrival appended to it before storage in the feedback database. The process controller transfers feedback data from the project site to the remote station via SMS. The SMS-to-database software subsystem is interrupted whenever a new feedback arrives so that the feedback can be collected and stored (as in table 2) for further processing.

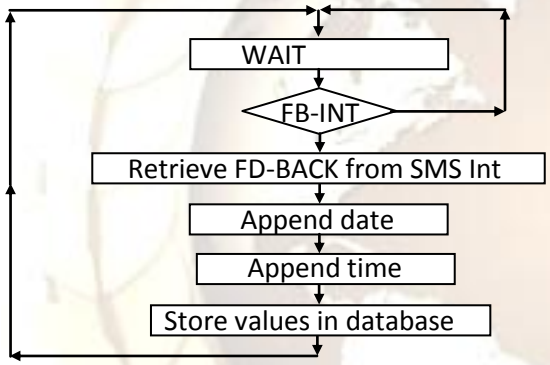

Fig 5 flowchart for SMS to database software subsystem

Table 2: Feedback database design

1.	Project ID	char [64]
2.	State code	Int
3.	Qualifier	int
4.	Actual-FDBK	Int (Hex)
5.	Date	dd mm yyyy
6.	Time	hh mm ss

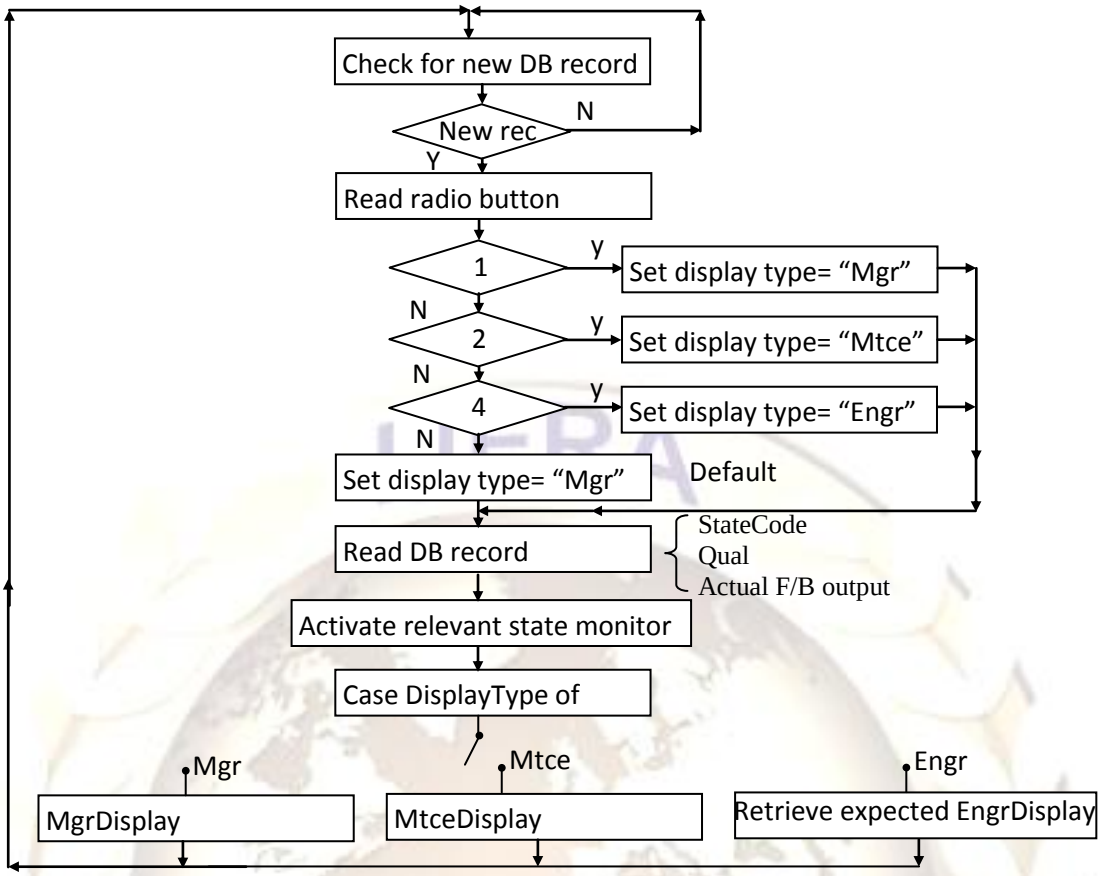

Fig 6: real-time remote process tracking software flowchart

Fig 6 shows the real time process tracking software flowchart. It uses the radio button to determine if the feedback is to be displayed for the manager, operator (or maintenance personnel) or for the engineer. The corresponding software captures feedback from an ongoing process and displays it for the relevant person as soon as it is received. The data can also be sent to the handset of the person intended, for greater convenience.

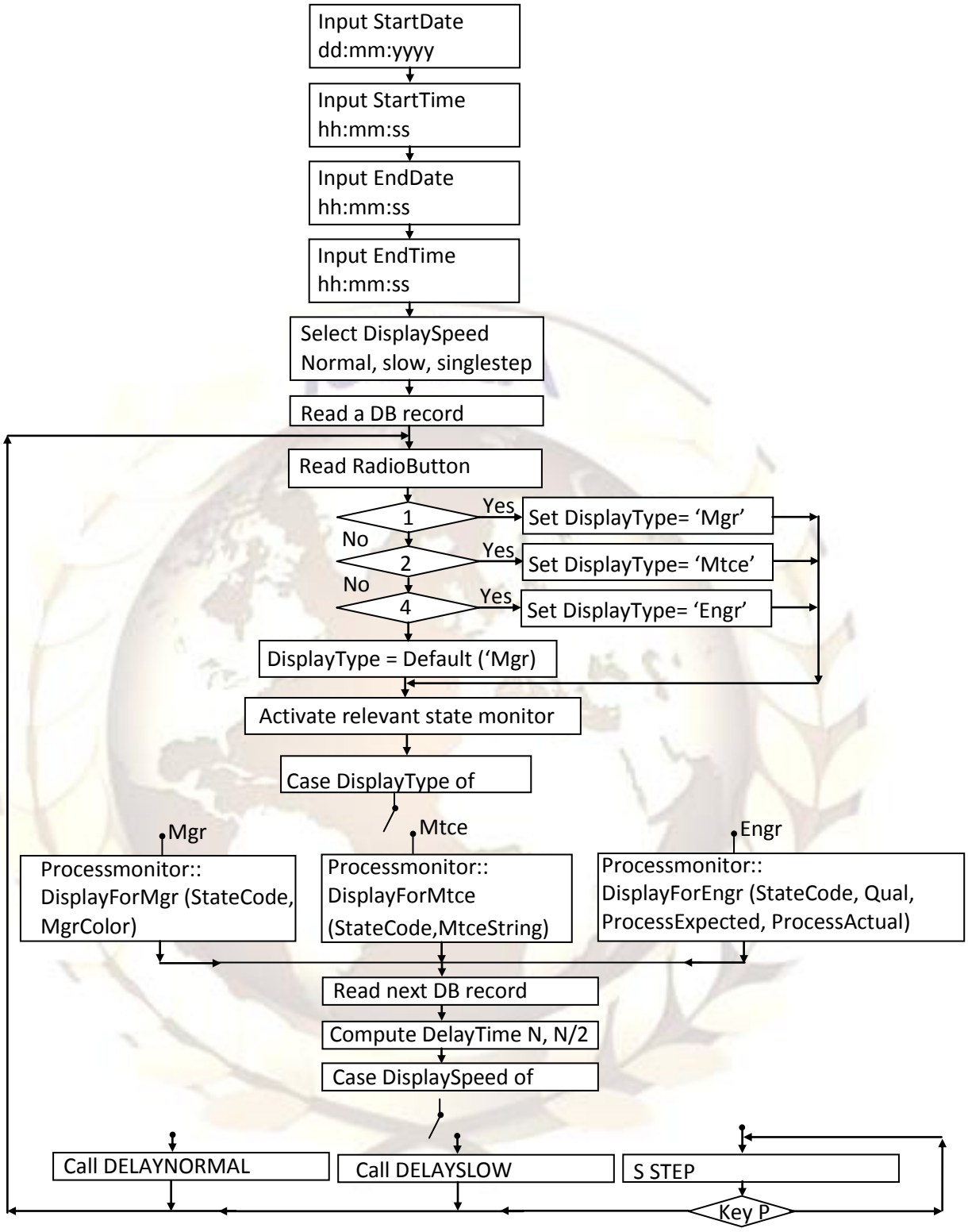

Fig 7: flowchart for process playback software

Fig. 7 shows the flow chart for the process playback software. It can display or play back the contents of the feedback database between two points in time as may be demanded by the user for further analysis of past machine cycles.

It can also

a. Display for manager
b. Display for operator or maintenance personnel
c. Display for the engineer

at the graphical user interface. The user has the opportunity to indicate how fast he or she wants the playback to be: real-life, slow motion or single step.

The remote process monitoring agent either activates the tracking software or the playback software depending on user choice. It then reads feedback data from the feedback database one record at a time and displays same on the GUI in a manner suggested by the setting of a set of 3 radio buttons of which only one may be active of any

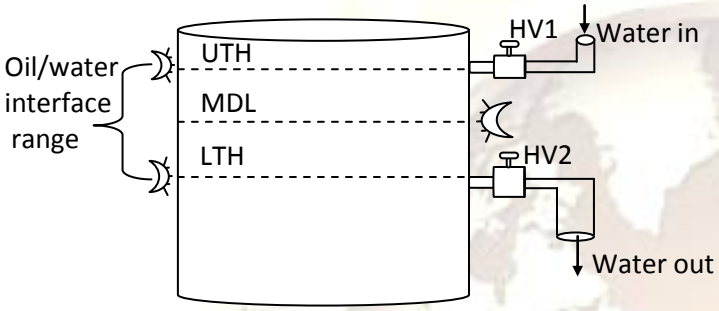

Fig 8: oil/water interface in an oil refinery

At the lower threshold level (LTH) the oil/water interface is considered too low. Therefore, water is pumped into the enclosure via the valve HV1 until the oil/water interface rises to middle (MDL) position. Thereafter, the dynamics of the process is allowed to take over and freely move oil/water interface up or down as determined by on-going processes.

At the upper threshold level (UTH) the oil/water interface is considered too high. So water is pumped out of the enclosure via HV2 until the oil/water interface falls to MDL. Thereafter the dynamics of the process is allowed to take over and freely move oil/water interface up/down as determined by the on-going processes.

given (see radio button definitions as part of fig 3). If the setting is for manager, the process monitor displays the feedback for manager. If the setting is for maintenance, the process monitor displays for the maintenance personnel or operator. If the setting is for the engineer, the process monitor displays diagnostic details to aid the engineer. The display setting via the radio buttons can be changed on the fly. The remote process monitor checks the radio button from record to record and uses the current setting for each record.

Typically, the manger, the engineer and the maintenance personnel are at different locations in the company. Each would have his or her own laptop or PC for monitoring. The same monitoring software runs on each of their machines but displays for each what his or her functionality demands. This is a great advantage in decision making.

V. Real Time Process Control Example

An industrial process example shall be used here to showcase remote process monitoring employing the Algorithmic State Machine (ASM chart) chart approach. Consider a tank used in the oil refinery industry (fig. 8) which contains both oil and water. The oil which is of lesser density is above the water. The oil/water interface level at any point in time is monitored.

At the threshold points (LTH and UTH), an alarm sounds to indicate unwanted condition. The alarm for oil/water interface too low sounds different from the alarm for oil/water interface too high. Both alarms would sound for 20 seconds and automatically stop. A red light indicator is used alongside the audible alarm to indicate UTH alarm while amber is used to indicate the LTH alarm. Green light indicator shows that the interface level is OKAY i.e. between LTH and UTH.

The algorithmic state machine chart representing the process of fig. 8 is shown in fig 9. At the decision boxes for LTH, UTH and MDL in the diagram, 0 represents water while 1 represents oil. The modified State Transition Table (STT), table 5, represents the information on the ASM chart (fig 9). This is then used to implement the oil/water level monitoring and control system as shown in fig. 10.

Table 3: Fully expanded table corresponding to fig 9

Link path	Present state name	Present state code B A	Qualifiers		Next state name	Next state code B′A′	State output			Conditional output		HEX Code for output
			LTH	Q			OKAY	HV2	HV1	LALTRG	HALTRG	
L1	ST0	00	0	0	ST2	10	1	0	0	0	1	51
L2	ST0	00	0	1	ST0	00	1	0	0	0	0	10
L3	ST0	00	1	0	ST1	01	1	0	0	1	0	32
L4	ST0	00	1	1	ST1	01	1	0	0	1	0	32

L5	ST1	01	0 0	ST0	00	0 1 0	0	0	08
L6	ST1	01	0 1	ST1	01	0 1 0	0	0	28
L7	ST1	01	1 0	ST0	00	0 1 0	0	0	08
L8	ST1	01	1 1	ST1	01	0 1 0	0	0	28
L9	ST2	10	0 0	ST2	10	0 0 1	0	0	44
L10	ST2	10	0 1	ST0	00	0 0 1	0	0	04
L11	ST2	10	1 0	ST2	10	0 0 1	0	0	44
L12	ST2	10	1 1	ST0	00	0 0 1	0	0	04

When a process control system is designed using the algorithmic state machine chart approach, it is often easier to represent a lot of information with just one byte of code and a table look-up technique. For example, in a state transition table (STT) (table 3), the present state code is concatenated with the qualifiers to form an address when the next state of the machine, the outputs for the present state and the conditional outputs are stored. Therefore, the table containing the next state code, state output and conditional output is stored at the remote site and the input byte (present state and qualifiers) and output byte (next state code and outputs) are captured from the project site and sent to the remote site every machine cycle. The address part of the feedback serves as an index into the table stored at the remote site to give fuller meaning to the present status of the machine.

For example, if the received code from the process side is 00 for BA and 100 for LTH, MDL and UTH (see first row of STT of table 3), this concatenated information reveals that the next state would be 10 for B′A′. The state output shows that the process is in its normal condition (i.e. OKAY) and the outputs from HV1 to HV2 are all at logic 0. This conveys clear concise information to the person at the remote end. Similarly, each concatenation of BA, LTH, MDL and UTH forms a code which points to a specific entry in table 3. Thus by arranging for such a code to be transmitted to the remote site along with the actual output byte for every machine cycle, it is possible to track the exact behavior of the process under control at the remote site. The data captured can be stored in the hard disk for future analysis of the process progress.

Fig 10 shows the block diagram implementation of the agent-based remote monitoring system used to monitor the for the oil/water interface control process. The process monitoring agent houses the process tracking software (of fig 6) and the process playback software (fig 7). When data is received from the remote site via the phone, the information is stored in the hard disk via the software data capture system whose flowchart is shown in fig. 5.

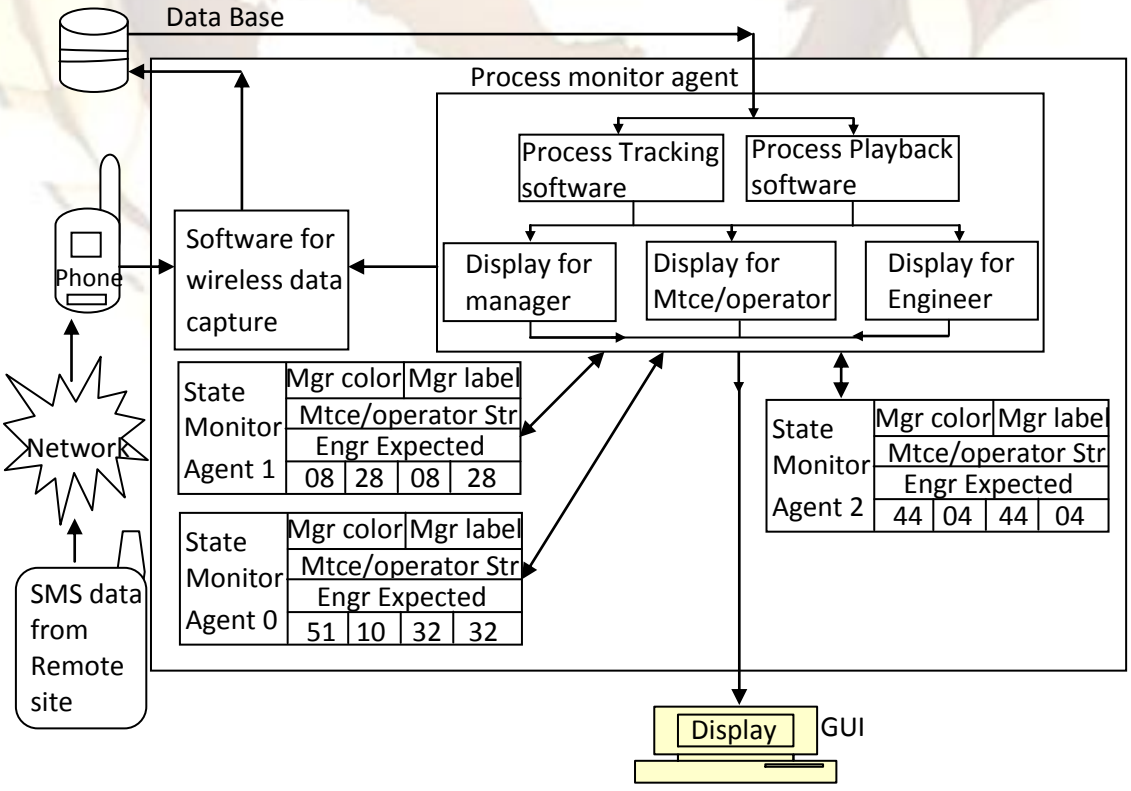

Fig 10: multi-agent based remote monitoring for oil/water interface

The process monitoring agent tracks the supplied information and sends it to the corresponding state monitoring agent to process. The state monitoring agent responds to the process monitoring agent and the information received is displayed on the GUI by the process monitoring agents for the stakeholders to view. It should be noted that these events happen fast as such one can choose to select the speed at which to view the information and also what time frame to view.

VI. Conclusion

Multi-agent based remote process monitoring has been ex-rayed in the foregoing. The method is generic and is not limited to monitoring agent-based process control systems. Other process control systems designed using any other method can be monitored using this method, except that the ASM chart and the modified STT must be provided to aid the design of the agent monitoring system. Again, this monitoring system is peculiar because of the way information is specified for all the personnel for easy viewing and understanding.

References

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury and A. T. Campbell, D. College. A survey of mobile phone sensing. *IEEE Communications Magazine*, 2010.

[2] M. Y. Ivory and M. A. Hearst, The state of the art in automating usability evaluation of user interfaces, *ACM Computing Surveys, 33(4),* 2001.

[3] D. Morley and C. S. Parker, *Understanding computers: today and tomorrow*. 10th edition. (Thomson Australia, 2005) Pp 9

[4] S. A. DeLoach, M. F. Wood and C. H. Sparkman, Multi-agent systems engineering, *International Journal of Software Engineering and Knowledge Engineering*. *11(3)*, 2001, 231-258.

[5] H.C. Inyiama, C. C. Okezie and I. C. Okafo, Agent based process control system design. *Proceedings of the peer reviewed 2012 national conference on infrastructural development and maintenance in the Nigerian environment. Nnamdi Azikiwe University, Awka, Nigeria.* 27-28 August, 2012. Pp 234-252.

[6] R. J. Tocci, N. S. Widmer, *Digital systems principles and application*, 8th edition, (Pearson Education, 2003). Pp 131-133

[7] H.C. Inyiama, C. C. Okezie and I. C. Okafo, Digital control of palm fruit processing using rom based linked state machines, *European Journal of Scientific Research, 59(4)*, 2011, p597

[8] Y. Peng, T. Finin, Y. Labrou, B. Chu, J. Long, W. J. Tolone, and A. Boughannam, A Multi-Agent System for Enterprise Integration, *International Journal of Agile Manufacturing*, 2(1), 1998, 213-229.

[9] K. P. Sycara, Multiagent Systems, *AI Magazine, 19(2*), 1998, Pp79-92.

[10] M. Wooldridge, M. Jennings, and D. Kinny, The Gaia methodology for agent-oriented analysis and design. *Autonomous Agent and Multi-Agent Systems, 3*, 1998, 285-312.