International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

Conceptual Design for Real-time Monitoring and Control System for critical assets using JAVA Classes: Case Study of Public Building and Oil & Gas Pipeline security

Engr. Fidelis C. Obodoeze[1], Dr. Ndidi Fidelia Ugwoke[2], Edith Angela Ugwu[3]

[1]Lecturer, Department of Computer Engineering Technology, Akanu Ibiam Federal Polytechnic, Unwana, Afikpo, Ebonyi State, Nigeria
[2]Senior Lecturer, Department of Computer Science Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria
[3]Lecturer, Department of Computer Science, Enugu State University of Science and Technology (ESUT), Enugu, Nigeria

ABSTRACT

Critical public assets such as oil and gas pipelines, public buildings, power plants, nuclear plants, public water utility etc needs to be monitored in real time so as to ensure their availability, dependability, safety and security. In this paper, we examined the basic components and architecture of real-time system, which includes real-time control, data acquisition system (DAS), critical systems, sensors and actuators, stimulus and response, Multi Agent System (MAS); design procedures and implementation of real-time systems using JAVA Standard Edition (SE) programming paradigm known as Java classes. We also examined the pros and cons of JAVA as a programming language compared to other languages such as Assembly Language and Ada. The paper went further to present the design and process architectures and timing requirements for two critical public assets- the oil and gas pipeline and public building involving different types of sensors and actuator interfaces. The paper concluded by implementing JAVA classes for the burglary alarm intruder detection system capable of detecting and reporting any intrusion targeted at critical public buildings.

Keywords: *real time, response, stimulus, JAVA classes, actuator, sensor, critical systems, Multi Agent System, data acquisition system, pipeline, public utility, building, alarm, alert, vandalism*

1.0 INTRODUCTION

Activities happening in oil and gas fields as well as other critical or important public assets such as buildings, bridges, water stations, power and nuclear plants need to be measured and monitored on real time to ensure their availability, dependability and security. Real time systems are systems that provide data processing and handling that are time-critical, i.e. if there is any delay in time for them to be processed and transmitted, there will be untold catastrophic consequences which may cause loss, harm, death or the degradation of environment. Monitoring and control systems are important classes of real-time systems. Real time systems check sensors providing information about the system's environment and *actuators* or *actors* take action when some exceptional sensor value is detected. Control systems continuously control hardware actuators depending on the value of associated sensors. The characteristic architecture of monitoring and control system is depicted in Fig.1.

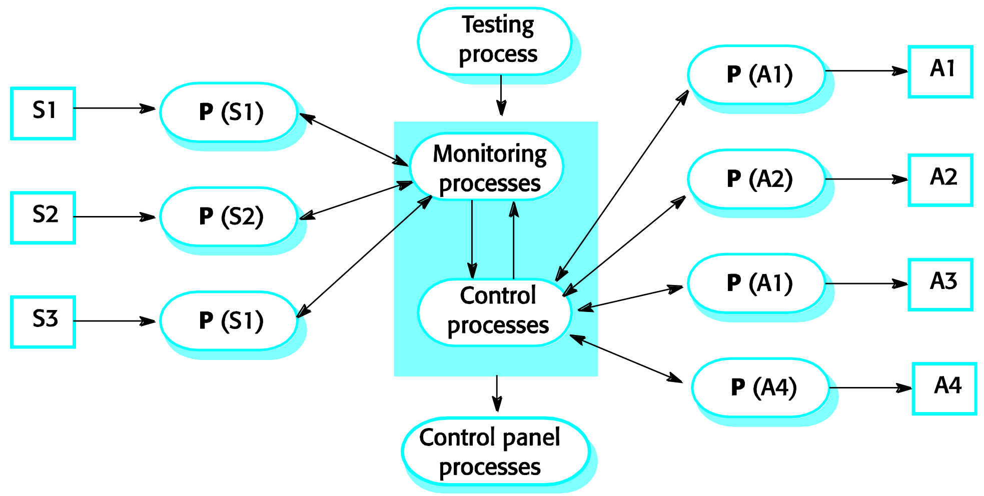

Fig.1. Generic architecture for a monitoring and control system [1]

S1, S2, S3 are sensors while A1, A2, A3, A4 are actuators. P(S1), P(S2) and P(S3) are sensor processes used for monitoring functions while P(A1), P(A2), P(A3), P(A4) are actuator processes used for control functions.

The sensor-actuator process is depicted in Fig.2. Here, the system controls several processes which may include: - 1. The system collects information from sensors, 2. It stores or buffers the information collected in response to a sensor stimulus, 3. It sends the information to a processor to carry out processing of collected information and computes the system response 4. The system generates control signals for the actuators or actors.

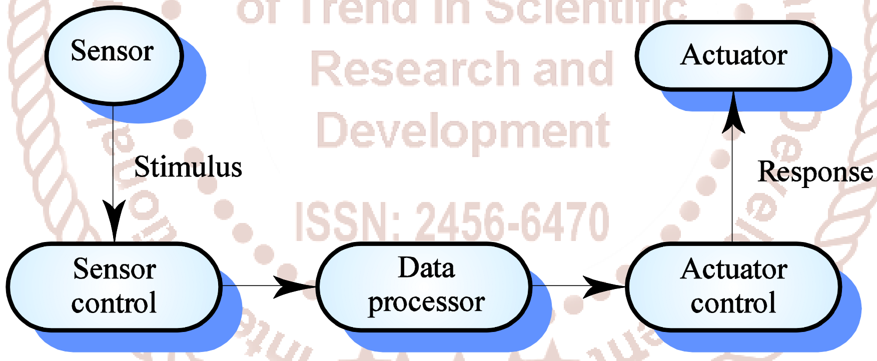

Fig.2. Sensor/Actuator processes [2]

A software sub-system is to be implemented to control a *vandalism alarm system* (a typical example of a real-time monitoring system) to monitor and control a critical asset such as an oil and gas pipeline system that carries raw crude oil, processed petroleum products such as Premium Motor Spirit (PMS) or Liquefied Natural Gas (LNG) or a critical facility such as a bridge or public building or any other facility that houses important national asset. The software system may have to be JAVA-based (Standard Edition, SE) while the hardware sub-system consists of several sensors/detectors and actuators. The sensors or detectors can be such that monitor or sense motion/movement, presence or hotspot, pressure or vibration etc. placed within the perimeter of the building or within the right-of-way (ROW) of the pipeline to detect the presence of vandals or intruders within the oil facility or asset. There can be up to fifty (50) or more of such sensors- motion, movement, presence or hotspot or/and pressure sensors depending on the location or the type of asset to be monitored or measured. The motion/movement or presence/hotspot sensors use some form of wireless signals such as Infra Red (IR) or Piezoelectric Infra Red (PIR) signals to detect its target.

Real-time monitoring and control system must guarantee system's dependability which includes availability, reliability, safety and security as depicted in Fig.3.

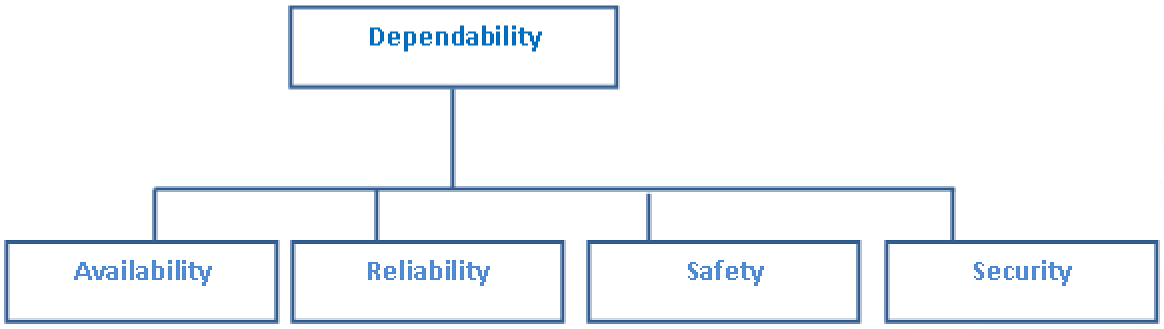

Fig.3. Dependability factor for real-time monitoring and control [3]

Availability is the ability of the system to deliver services when requested or needed. Reliability is the ability of the system to deliver services as specified while safety is the ability of the system to operate without catastrophic failure. Security is the ability of the system to protect itself against accidental or deliberate intrusion.

1.1 Timing requirements of a stimulus/response pair

The number of sensors to be polled or measured and the timing requirements of the system are used to calculate how often each process has to be scheduled. For example, the movement or motion detectors process must run 500 times per seconds (i.e. 500Hz) because there may be up to 250 movement/motion sensors or detectors in the system.

Similarly, there can be up to 100 pipeline IR hotspot detectors or sensors that must be checked or polled twice per second.

Likewise, the pressure sensors installed along the right of way (ROW) of the pipelines can be up to 100 in number; and these 100 pressure sensors need to be monitored or polled every second (1 Hz).

Table 1 depicts some typical stimulus/response timing requirements required for the design of real-time monitoring and control of critical public asset such as oil and gas pipeline, public building, power plant, water station, nuclear plant or bridge.

Table 1. Stimulus/response timing requirements

	Stimulus/Response	*Timing Requirements*
1	Movement/Motion detector or sensor	Each movement/motion detector or sensor should be polled twice per second
2.	Communications	The call or SMS alert to the Police, JTF, or Civil Defence officials or nearby local vigilante guards should be started within 2 seconds of an alarm being raised by a sensor or detector.
3	GSM alert/Call Synthesizer	A synthesized message should be available within 4 seconds of an alarm being raised by a sensor. The synthesizer message can be a GSM call or SMS alert to the nearby local Police, Vigilante or Civil Defence patrol officials.
4.	Audible Alarm	The audible alarm should be switched on within 1-2 seconds of an alarm being raised by any of the sensors
5.	Door Alarm	Each door alarm should b polled twice per second.
6.	Window Alarm	Each window alarm should b polled twice per second.
7.	Light switch	The lights should be switched on 1-2 seconds an alarm is being raised by a sensor.
8.	Power fail interrupt	The power switch to backup power must be completed within a deadline of 50 seconds deadline.

Fig.4 demonstrates the implementation of the process architecture and timing requirements of a real-time vandal/intruder monitoring and alert system for an oil and gas pipeline monitor. Here in the diagram, the line associated with each process on the top-left is used to indicate how the process is controlled (i.e. dotted lines). The lines on a periodic process are solid lines with the minimum number of times a process should be executed per second as an annotation. A periodic process has dashed lines on their top-left corners, which are annotated with the event that causes the process to be scheduled or actuated. The sensors required to monitor the presence of an intruder or vandal are hotspot detector, motion/movement sensor and vibration and pressure sensors are used to confirm that vandalisation of a pipeline has taken place.

Fig. 5 depicts the processes for a burglary alarm system to monitor a public building to detect an intruder. The burglary alarm system for a public building is a system that is required to monitor sensors on doors and windows to detect the presence of intruders in a building. When a sensor indicates a break-in, the system switches on lights around the area and calls police automatically. The system should include provision for operation without a mains power supply such as solar or inverter power backup.

The sensors for burglary alarm system may include:- Movement detectors, window sensors, door sensors; 50 window sensors, 30 door sensors and 200 movement detectors; Voltage drop sensor.

The Actions to be taken by the actuator may include:-

1. When an intruder is detected, police are called automatically;
2. Lights are switched on in rooms with active sensors;
3. An audible alarm is switched on;
4. The system switches automatically to backup power with solar or inverter power system when a voltage drop is detected.

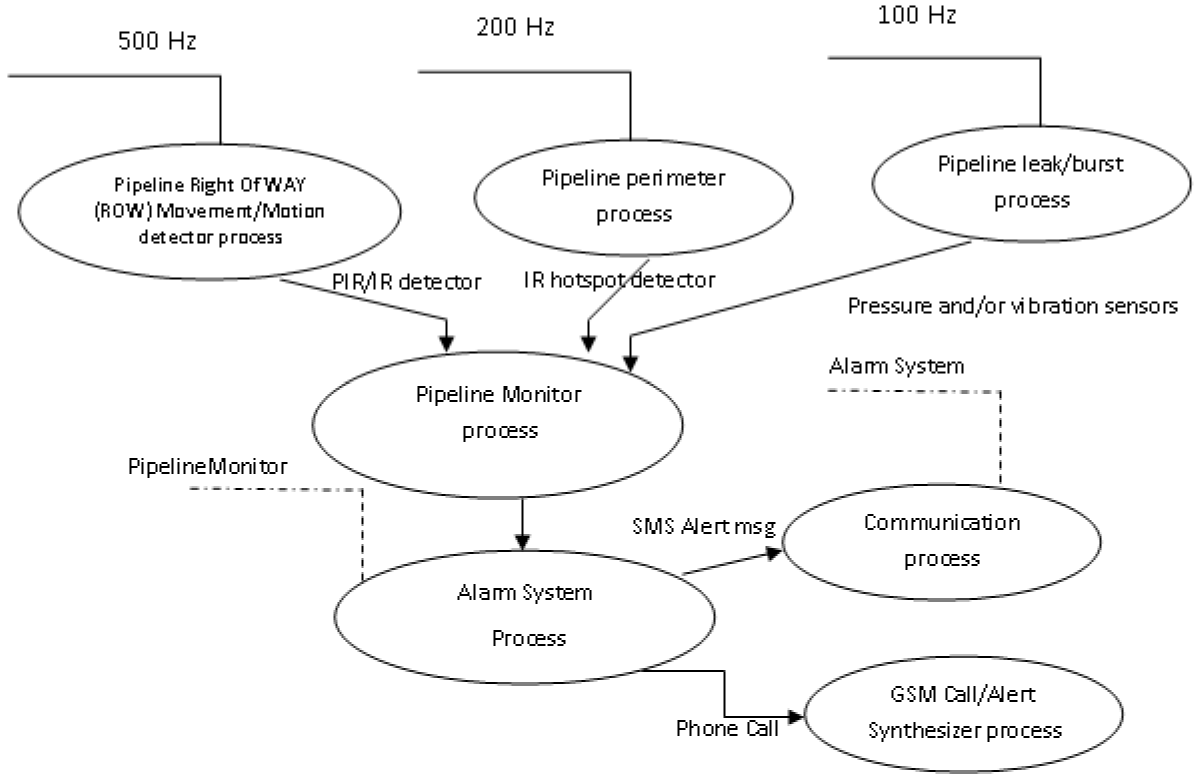
500 Hz
200 Hz
100 Hz
Pipeline Right Of WAY (ROW) Movement/Motion detector process
Pipeline perimeter process
Pipeline leak/burst process
PIR/IR detector
IR hotspot detector
Pressure and/or vibration sensors
Alarm System
Pipeline Monitor process
PipelineMonitor
SMS Alert msg
Communication process
Alarm System Process
Phone Call
GSM Call/Alert Synthesizer process

Fig. 4: Process architecture of the vandal/intruder pipeline monitoring and alert system

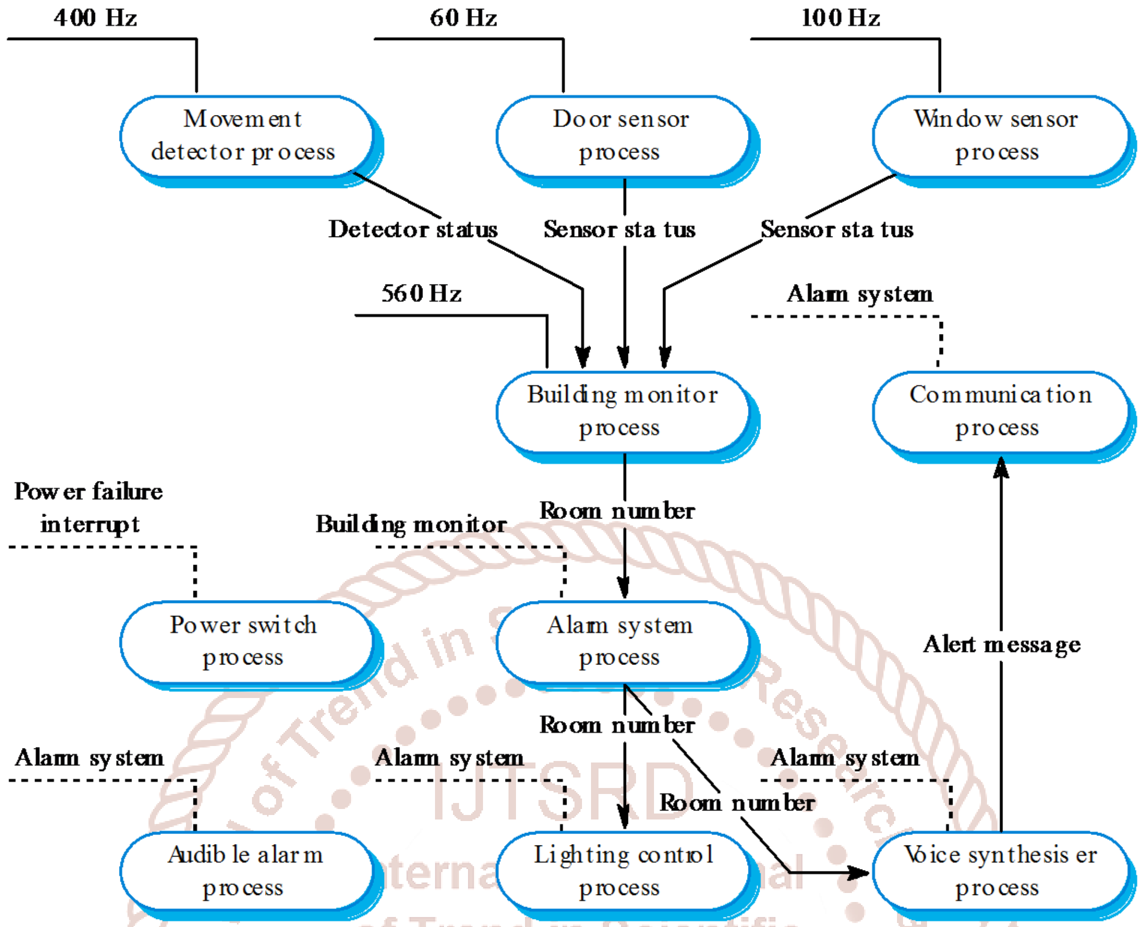

Fig. 5: Process architecture of the burglary alarm monitoring system for a public building [4]

1.2 Critical Systems

System failures are relatively common. In most cases, these failures cause inconveniencies but no serious, long-term damage. However, in some system failures can result in significant economic losses, physical damage, environmental degradation or threats to human life. These systems are called *critical systems*.

Critical systems are technical or socio-technical systems that people or businesses depend on. If these systems fail to deliver their services as expected, then serious problems and significant looses may result. For example, if an oil pipeline supplying an oil depot or refinery from an oil well is vandalized or ruptured by vandals or criminals, not only will be economic loss be incurred due to loss of the product revenue but the environment will be degraded or even loss of human lives may occur. Another example is vandalism of power plant, electric transform or high-tension cables supplying electric power to industrial and residential customers. This can result to loss of energy (power) to important customers and millions of people may be thrown into darkness; and industries may not produce except they have a backup alternative (which is very costly to run and maintain); damaged high-tension cables can electrocute human beings and lead to loss of precious human lives. Critical systems, therefore, need to be monitored and controlled using real-time systems to ensure dependability at all times.

- A burglar alarm system is primarily a monitoring system. It collects data from sensors but no real-time actuator control
- Control systems are similar but, in response to sensor values, the system sends control signals to actuators

An example of a monitoring and control system is a system which monitors temperature and switches heaters on and off. Fig.6 depicts a temperature control system and its process architecture and timing requirements.

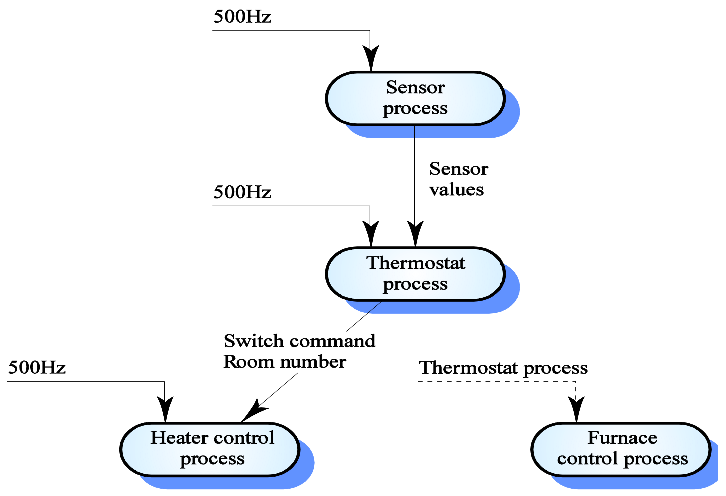

Fig.6. A temperature control system and the process architecture [5]

A burglary alarm and control system is a system that is required to monitor sensors on doors and windows to detect the presence of intruders in a building when a sensor indicates a break-in, the system switches on lights around the area and calls police automatically. The system should include provision for operation without a mains power supply such as a solar-powered backup or inverter system.

1.3 Data acquisition systems (DAS)

Data acquisition system (DAS) is an essential component or part of a real-time monitoring and control system. DAS accomplished the following objective in a real time monitoring and control system:

- *It collects data from sensors for subsequent processing and analysis.*

Fig.7 shows a typical Data Acquisition System (DAS) with six (6) sensors S1, S2, S3, S4, S5 and S6. The data acquisition starts with of the sensor process, i.e. sensor identification and its corresponding value, then to storing of the sensor data in buffer and thereafter the processing of the sensor data and finally the display or communication of the sensor message.

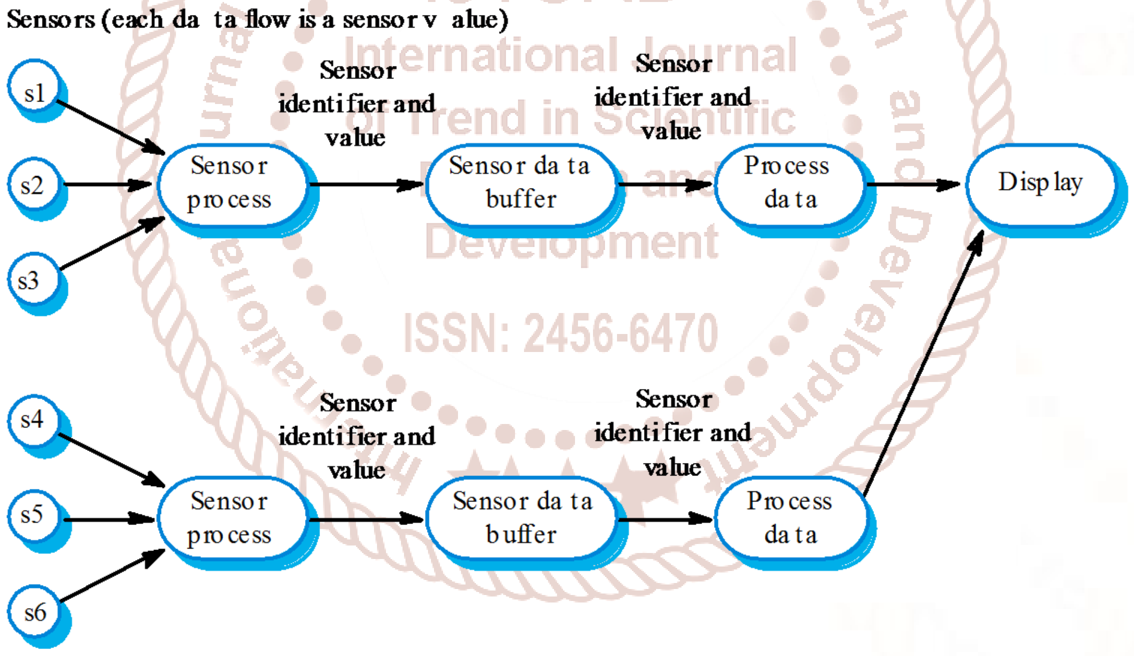
Sensors (each da ta flow is a sensor v alue)
s1
s2
s3
Sensor identifier and value
Sensor identifier and value
Sensor process
Sensor da ta buffer
Process da ta
Disp lay
s4
s5
s6
Sensor identifier and value
Sensor identifier and value
Sensor process
Sensor da ta buffer
Process da ta

Fig.7. The Data Acquisition process [6]

2.0 Java as a real-time programming language

Java is a versatile programming language that supports some form of real-time programming and security monitoring; it supports lightweight concurrency (threads and synchronized methods) and can be used for some soft real-time systems. According to [7], the use of Java Standard Edition (SE) APIs in the implementation of real-time systems is most appropriate for soft real-time development. Using Java SE for hard real-time development is also possible, but generally requires the use of more specialized techniques such as the use of *NoHeapRealtimeThread* abstractions, as described in the Real-Time Specification for Java (JSR 1), or the use of the somewhat simpler *ManagedSchedulable* abstractions of the Safety Critical Java Technology specification (JSR 302).

It is also important to distinguish *real-time engineering*, as it is described in this series, from *performance engineering*. An e-commerce web server, for example, might have been carefully engineered to support an average of 1,000 transactions per second. That is different from saying that every transaction must be completed in 1ms. It could be that some transactions require hundreds of ms and others are completed in less than 1 ms, as long as the average of all transactions is 1ms. It could also mean that each transaction requires an average of 4 ms from start to end, but the system has the ability to concurrently execute four transactions at a time.

The benefits of the Java language are especially valuable in real-time applications that are large, complex, and dynamic. Software engineers are motivated to select Java SE when their projects require dynamic code updates, coordination between multiple teams of developers, integration of independently developed software components, support for multiple hardware or operating system platforms, or support for multiple software configurations as product requirements evolve over multiple years or even decades.

Projects that can be implemented entirely by one or two developers in a year's time are more likely to be implemented in a less powerful language such as C or C++, especially if it is critical that the end product minimize consumption of battery power or high-volume production costs. Such projects are less likely to appreciate the benefits of Java, and are often able to justify the higher software engineering and maintenance costs associated with the choice to use an older language.

The following points summarises the characteristics of Java programming language:

i. Java supports lightweight concurrency (threads and synchronized methods) and can be used for some soft real-time systems
ii. Using Java SE for hard real-time development is also possible but requires some special implementation using *NoHeapRealtimeThread* abstractions or *ManagedSchedulable* abstractions.
iii. Java 2.0 is not suitable for hard RT programming or programming where precise control of timing is required
   - *Not possible to specify thread execution time*
   - *Uncontrollable garbage collection*
   - *Not possible to discover queue sizes for shared resources*
   - *Not possible to do space or timing analysis*

2.0.1 Real-time programming

Hard-real time systems may have to be programmed in assembly language to ensure that deadlines are met. Languages such as C allow efficient programs to be written but do not have constructs to support concurrency or shared resource management. Ada as a language designed to support real-time systems design so includes a general purpose concurrency mechanism.

Fig.8 depicts the real-time system model.

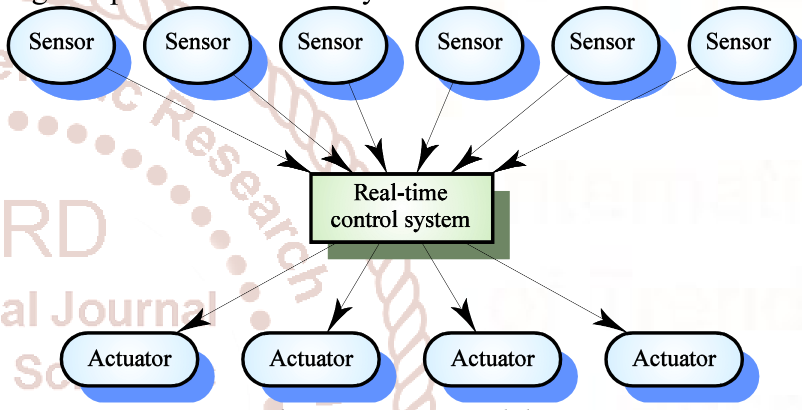

Fig 8. Real-time system model[8]

3.0 Software Design Methodology

The real-time programming design methodology that is very suitable for programming multiple sensors and actuators (actors) is Multi Agent System (MAS) design methodology. The following section describes this design methodology.

3.0.1 Multi Agent System (MAS)

Multi Agent intelligent systems (MAS) enable multiple software agents to act and collaborate together to monitor a critical facility on real-time and enables actuators to take action in the case of a breach, attack or vandalism of a critical facility.

The following methods execute the sense-deliberate-act control when an event is detected by the sensor from its environment.

```
Public void execute{
// 1. Sense environment
SenseEnvironment ();
// 2. Deliberate…….
//3. Act: Select one or more activities:
For (Predicate activity: activities)
 performActivity (activity);
```

3.0.2 Environment Interface
The interface between an agent and its environment is based around two core components: sensors and actions.

Sensors are responsible for generating the agents model of its environment while actions are components that cause some change to occur in the environment. The core focus of sensor is belief generation and the core focus of an action is to facilitate manipulation of the environment.

3.03 Objectives of a real-time system software design
The objectives of real time software design include the following:-

1. To explain the concept of a real-time system and why these systems are usually implemented as concurrent processes
2. To describe a design process for real-time systems
3. To explain the role of a real-time executive
4. To introduce generic architectures for monitoring and control and data acquisition systems

4.0 Hardware and Software Design and Implementation
The following steps should be taken to realize a dependable real-time system design and implementation.

Step 1: Establish system requirements

Step 2: Partition the requirements along hardware and software requirements

Step 3: Handle the hardware design and software design respectively

The following section describes the hardware and software design methodologies employed in the design of pipeline and burglar alarm and control systems.

4.0.1 Real-time Design processes
The following are the steps to be taken to design the real-time processes for the pipeline monitoring system and burglary alarm monitoring system for a public building:-

1. *Identify stimuli and associated responses.*
2. *Define the timing constraints associated with each stimulus and response.*
3. *Allocate system functions to concurrent processes.*
4. *Design algorithms for stimulus processing and response generation.*
5. *Design a scheduling system which ensures that processes will always be scheduled to meet their deadlines.*

4.0.2 Simulation of the Response Stimuli
The stimuli to be simulated for the pipeline monitoring system and burglary alarm system for a public building are both: 1. Power failure and, 2. Intruder alarm 3. Pressure or vibration sensors

1. Power failure stimuli are generated a periodically by a circuit monitor. When received, the system must switch to backup power (solar powered backup) within 50 ms.

2. Intruder alarm Stimulus is generated by system sensors. Response is to call the police or nearby security patrol guards, switch on building lights and the audible alarm.

3. Pressure or vibration sensors are simulated to confirm the existence of a vandalisation in pipeline architecture along its right of way (ROW). This confirms to the security officials that indeed vandalism has taken place.

4.1 JAVA classes implementation for Burglary Alarm Intruder detection system for Public Building
The JAVA classes or source codes are depicted below:

```
class BuildingMonitor extends Thread {
                BuildingSensor win, door, move ;
                Siren     siren = new Siren () ;
                Lights    lights = new Lights () ;
                Synthesizer synthesizer = new Synthesizer () ;
                DoorSensors doors = new DoorSensors (30) ;
                WindowSensors windows = new WindowSensors (50) ;
                MovementSensors movements = new MovementSensors (200) ;
                PowerMonitor pm = new PowerMonitor () ;
                BuildingMonitor()
```

```
        {
                // initialise all the sensors and start the processes
                siren.start () ; lights.start () ;
                synthesizer.start () ; windows.start () ;
                doors.start () ; movements.start () ; pm.start () ;
        }

public void run ()
{
        int room = 0 ;
        while (true)
                {
                        // poll the movement sensors at least twice per second (400 Hz)
                        move = movements.getVal () ;
                        // poll the window sensors at least twice/second (100 Hz)
                        win = windows.getVal () ;
                        // poll the door sensors at least twice per second (60 Hz)
                        door = doors.getVal () ;
                        if (move.sensorVal == 1 | door.sensorVal == 1 | win.sensorVal == 1)
                                {
                                        // a sensor has indicated an intruder
                                        if (move.sensorVal == 1)      room = move.room ;
                                        if (door.sensorVal == 1)      room = door.room ;
                                        if (win.sensorVal == 1 )      room = win.room ;

                                        lights.on (room) ; siren.on () ; synthesizer.on (room) ;
                                        break ;
                                }       }
                lights.shutdown () ; siren.shutdown () ; synthesizer.shutdown () ;
                windows.shutdown () ; doors.shutdown () ; movements.shutdown () ;

        } // run
}                                                                                    //BuildingMonitor
```

5.0 SUMMARY AND CONCLUSION

This paper presented the architecture and components of real-time systems that are required to monitor and control critical systems to ensure their dependability. The paper also presented two critical systems- oil and gas pipeline and public building as case studies and showcased their hardware and software requirements as well timing diagrams.. It also highlighted the components of real-time system and pros and cons of JAVA in the implementation of real-time design, control and monitoring using JAVA classes. Real-time systems are designed in such a way that they will respond to events that occur with time so that there will not be system failure. JAVA classes can be used to design the real-time software and hardware control functions to monitor real-time critical systems. JAVA is versatile and has been deployed in designing real-time system programming handling functions such as design, abstractions, control and security monitoring.

REFERENCES

1. Ian Somerville, 2007. Monitoring and Control Systems. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.349-350.
2. Ian Somerville, 2007. System Design. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.342-343.
3. Ian Somerville, 2007. System Dependability. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.47-48.
4. Ian Somerville, 2007. Real-time software design. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.349-352.
5. Ian Somerville, 2007. Monitoring and Control System. Software Engineering 8th Edition.

Addison-Wesley Publishers Limited London. Pp.354.

6. Ian Somerville, 2007. Monitoring and Control System. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.355.
7. Kelvin Nilsen, 2014. " Developing Real-Time Software with Java SE APIs: Part 1". Accessed online on 12th March 2018 at http://www.oracle.com/technetwork/articles/apd/real-time systems using JAVA Classes/Developing Real- Time Software with Java SE APIs_ Part 1.html
8. Ian Somerville, 2007. Real-time software design. Software Engineering 8th Edition. Addison-Wesley Publishers Limited London. Pp.341