Research Paper

On

**AUTOMATION AND EXCEPTION HANDLING IN MISSION CRITICAL SYSTEM**

Author

Vignesh Kumar Karthikeyan Rajalakshmi

A20424508

Professor

Dennis Hood

**Abstract**

* **Introduction:**
* **Mission-Critical Systems**
* What is a mission-critical system?
* How do you deem a system to be mission critical?
* **Automation**
  + What is an Automation?
  + Impacts of Automation
* **Exception Handling**
  + Introduction
  + Types of Exception handling

2.1 Hardware Exception handling

2.2 Software Exception handling

* + Importance of Exception handling
* **Overview:**
* **Automation in Mission-Critical system**
* Necessity of automation in mission-critical system
* Challenges of implementing automation
* How to calculate the efficiency of automating
* Benefits of automating mission critical systems
* **Exception handling in Mission-Critical system**
* Need for exception handling in mission-critical system
* Performance of Exception handling in automated MC systems
* Benefits of using Exception handling in MC systems
* **Design and testing for Mission-critical system**
* Designing a Mission-critical system
* Testing and Quality Assurance
* Automation and Exception handling in design and testing
* **Case study**
* **Case study: Chernobyl**
* Automation in case study
* Exception handling in case study
* **Conclusion**
* **Future Scope**
* **Bibliography**

***Abstract***

In our daily life, we come across several mission-critical systems such as banking transactions, power supply, and satellite communications. These systems are efficient and critical for individual people to large organizations. Here, we discuss how to automate these functions or processes and make them enough fault-tolerant to prevent any disastrous situation from occurring. Automating a mission-critical system might be seen as handing over an important task to a complete stranger but this improves efficiency and makes it less prone to human error.

**Introduction**

**Mission-Critical System**

**What is a mission-critical system?**

The mission-critical system is generally a system deemed to be a core structure of a firm or organization whose functionality impacts the particular organization greatly. These are present in our day to day life such as Electricity transmission, aircraft, and railway traffic, hospitals and also in advanced scientific technologies such as space crafts, satellites, and nuclear power plants.

**What deems a system to be mission critical?**

A system that controls the fate of a product’s functionality such as production, transaction, output control, quality maintenance is categorized as a Mission-Critical (MC) system. Any negative impact in this system could lead to the disaster of the company. If this system is shut down or made useless in any means then it will completely break apart the firm or organization to which the system belongs to.

**Automation**

**What is an Automation?**

Automation can be defined as performing a task with minimal or no manual interaction. Automation can be done for small tasks such as messaging, ticketing, opening and closing of doors to huge and complex tasks such as driving, air traffic control, piloting, production.

**Impacts of Automation**

“Automation”, this term has seen some ups and downs throughout the world. Automating some complex tasks has led to a reduction in time and the cost to produce the product. This is true in the case of software engineering tasks such as coding, designing, and testing. They also provide consistency in the tasks with less or no errors at all. Automating complex tasks can be difficult since it needs more training, algorithms, and steps to be followed. But, this exposes them to less error-prone environments and a reduction in human errors.

Automation is viewed as a tool that put humans out of jobs. In many of the domains people are replaced with machines and AI (Artificial Intelligence) systems. This can be argued with cost reduction and human life preservation such as automatic piloting in drones and space crafts.

**Exception handling**

**What is Exception handling?**

Handling of events that are unexpected to happen is called exception handling. Exception handling is used most widely in the programming world to prevent any situation that is considered to be bad from happening. They are also a safety mechanism to prevent the system from falling into an irreversible state.

Types of exception handling

Based on the detail of how and where the exception is being handled they can be classified into types. A general classification of exception handling is

1. Hardware exception handling
2. Software exception handling.

**Hardware Exception handling**

Hardware exceptions can be handled smoothly by halting the tasks and resuming them later. If a product has both the software and hardware then when there is an error that cannot be avoided through the software system can be handled through hardware exception handling. Example: A processor in a computer can halt the task if there is an error due to some software.

**Software Exception handling**

Software exception handling is widely used across several software platforms. While writing code for the software product there can be many functionalities that need to be implemented which can lead to different outputs based on the input and the action performed. Sometimes, the unexpected action can cause crashing of the product and sometimes it can even corrupt the file too. This can be prevented by giving some handlers for these unexpected outputs.

**Importance of Exception handling**

Exception handling is considered as important in the hardware and the software level due to its massive use of preventing the system from being pushed into looping or irreversible state. The system of the product may not know how to react when it receives the command which it is not exposed to. During these cases, there has to be a failsafe mechanism that takes the state into custody and provide a temporary solution. This can be done with the help of Exception handling. There is also the method of error checking, a replacement for exception handling is used but sometimes without knowing the type of error this method cannot be implemented. This exception handling provides the best-case scenario for usage and safety provision.

**Overview**

**Automation in Mission Critical System**

**Necessity of Automation in Mission-Critical System**

Automation in a mission-critical system is required because of the possibility that it can help in the prevention of risking human life. Example: In the case of the auto-piloted drones, even in the extreme cases where the plane can be shot down by enemy defenses it helps in saving a human life which is priceless when compared to the drone. This can be said for Nuclear reactors as well as other space missions.

**Challenges of Implementing Automation**

Automating a system is not a simple task that can be achieved overnight. It takes patience and a lot of time to train and test the results of each change. Automating a simple task itself needs so much time for it to be successful. Consider automating a mission-critical system that a whole organization relies on. It has to be perfect or else it will lead to error and disaster for the company. The cost that it takes to implement automation for a task should not exceed the limit that it takes for human labor who works for a certain period.

**How to calculate the efficiency of automating**

The efficiency of automation can be calculated based on its capability to reduce the cost, time taken, error occurrence while performing tasks or operation. It also includes the ability to increase the production rate or producing with consistent results.

**Benefits of automating a mission critical system**

When the challenges are faced and overcome then the benefits obtained by the automation can be immense. Increase in production efficiency, cost reduction, reduced time consumption, avoid risking human life (auto-pilot drones, nuclear reactors, space missions, etc.) There would be consistency in the output when a task is automated by a machine rather by human intervention.

**Exception handling in Mission-Critical system**

**Need for Exception handling in Mission Critical system**

Exception handling has proven to be the best method for managing the unexpected events that occur during a task. This needs to be present in the mission-critical system since it has the ability to prevent the MC from crashing or intruded with numerous bugs.

**Performance of Exception handling in MC systems**

Performance of an exception handling can be measured only by means of testing it by subjecting the system under various conditions. If it is software then it can be tested with test cases of different inputs that the system might not expect and see how it reacts. If it is hardware then it has either halt its task when something unexpected happens or it should let know the responsible person to proceed further.

**Benefits of Exception handling in automated MC systems.**

The benefit of having Exception handling in an MC system can prevent an organization or nation from creating a disaster based on the task it performs. This can provide a fail-safe mechanism for the task to be safely executed or to be halted until further notice to proceed is given. This reduces the cost of repairing and also reduces the error from happening.

**Designing and Testing for Mission-critical system**

**Designing a Mission-critical system**

* **What is Design?**

Design is a structural representation of how a system will look like and how a system might function. This gives an overall idea for the users about the workflow and responsibility of each component in a system

Designing a mission-critical system takes more effort than designing a normal system. The flow of work is more important in the MCC systems. Hence to portray a design for a critical system it needs to be accurate and it shouldn’t mess up or confuse any succeeding steps on the implementation. The better the design with less complexity then it will be easy for the actual development of the system. Reducing the complexity shouldn’t affect the clarity of the system’s functions

**Test case and Quality Assurance for MC systems**

* **What is a test case?**

Test cases are steps that check whether the system performs as it is intended to perform. This is an important step where each nook and crook of a product or a system is tested out to find any possible errors or exceptions.

**Quality Assurance**

Quality assurance is a method to make sure that the system performs to its fullest and doesn’t meet with any error. The end product delivered should be satisfying with its look and performance. Mainly, the product shouldn’t compromise its quality with its look or structure.

**Exception handling and automation in design and testing**

A mission-critical system can be automated by sampling it with other mission-critical systems which might be similar. Though the major implementations cannot be trusted with the automation, the initial and simple structural designs can be automated and thus it reduces the time and helps in increasing the focus of designers on more complex ones.

Cases sometimes fail and then they might cause the system to be in an infinite loop. To counter this, the test case result can be put through exception handling which might mitigate these minor issues and helps in handling the system effectively. They can also help in understanding where the error might happen and can help fix the problem. Test cases can even be automated as same as the design. Simpler test cases automation can reduce time wastage. Automation and exception handling has to co-exist in this kind of environment which would further increase the efficiency of the system.

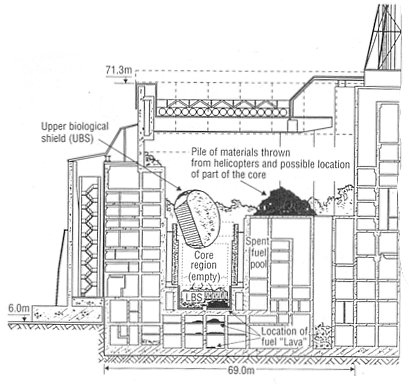
**Case Study**

The case study for Automating and Exception handling for the Mission-critical System I have taken is **Nuclear reactors** **and Power plants**. Here we discuss how the various safety and prevention mechanisms provided in the nuclear power plant can be done with less or without the human intervention. We will discuss about the benefits of having automation and exception handling and will justify why it has more pros than the cons.

**Case study: Chernobyl**

The Chernobyl incident happened in a nuclear reactor in Ukraine due to fault in the structure of the power plant and also due to inadequate operation by the personnel. This has led to 28 deaths due to the result of **ARS** (Acute radiation syndrome). Reactor design itself is unique and hence it doesn’t need to be feared about the incident such as this happening around the world.

Cause of the disaster as per the **world-nuclear.org** is due to some tests performed without proper precautions taken. To improve the turbine capacity of the reactor they have put new voltage regulators to stabilize it. And the operator has done so many tests on it which includes even disabling the reactor’s automatic shutdown system. This caused the control rods to overheat while they were being immersed in the reactor. This caused huge pressure to occur inside the reactor and hence they resulted in damaging and releasing lots of steam inside. This caused the burst of the reactor and releasing all the radioactive materials into the atmosphere.



**Fig: 1.** The damaged Chernobyl unit 4 reactor building

**Automation in Case study**

The automation, in this case, would be where the testing of all the components that are necessary for the functioning of the nuclear power plant. These prevent any manual fault that causes damage to the reactor which incase can lead to utter disaster. In this case, the turbine capacity which has been tested manually could have been done automatically. The nuclear reactor’s automatic shutdown is possible and that’s why other inferior features can be easily implemented.

**Exception handling in case study**

Exception handling would have been an amazing use in this case. The control rods getting overheated while it was being inserted caused the steam and pressure inside the reactor. This could be prevented if this was handled by the system. The reactor should have shut down after seeing the situation that has caused some irregularity. Which in the case could have prevented this final disaster. The second case is where when there was pressure inside the reactor was too high the system would have deducted and released these pressurized air through some exit valve.

**Conclusion**

Based on all the above-discussed topic we come to see the advantages of having Automation and the Exception handling and its efficiency. Hence we arrive at the conclusion that these two are required features in any of the mission-critical systems

**Future Scope**

The Future of this topic has enough room for improvement. It can include Artificial Intelligence (AI) and Machine Learning (ML) to be implemented. Once they are done then even if some extreme cases happen then these AI can learn from the tasks performed repeatedly and take necessary measures to avoid any kind of losses. ML can be helped in predicting what can happen if the current state continues and what steps can be helpful in preventing a situation from occurring.

As AI and ML are much advanced and there are lot of potential engineers around the world with proficient scientists who can give tons of data on nuclear materials and reactions. These are not some random data. These are data inferred, observed and verified through lots of practice, test and experiments on different materials and hence this holds an absolute proof of what happens in each scenario. A nuclear power plant cannot be built with some good civil engineering capability it also needs scientific knowledge about the materials they are going to use and what kind of effects would be there if they used wrong elements and stuff. Since they have achieved to contain the dangerous material inside a building and putting it for a good use then there is a lot more chance for the humankind to prevent any harm occurring from it.

**Bibliography**

[1] Chong, Jennifer, Partha Pal, Michael Atigetchi, Paul Rubel, and Franklin Webber. "Survivability architecture of a mission critical system: The DPASA example." In *21st Annual Computer Security Applications Conference (ACSAC'05)*, pp. 10-pp. IEEE, 2005.

[2] Wissink, Tom, and Carlos Amaro. "Successful test automation for software maintenance." In *2006 22nd IEEE International Conference on Software Maintenance*, pp. 265-266. IEEE, 2006.

[3] Lippert, Martin, and Cristina Videira Lopes. "A study on exception detection and handling using aspect-oriented programming." In *Proceedings of the 22nd international conference on Software engineering*, pp. 418-427. ACM, 2000.

[4] Teller, Edward, Muriel Ishikawa, and Lowell Wood. *Completely automated nuclear reactors for long-term operation*. No. UCRL-JC--122708. Lawrence Livermore National Lab., 1996.

[5] Hecht, Myron, Jeffrey Agron, Herbert Hecht, and K. H. Kim. "A Distributed Fault Tolerant Architecture for Nuclear Reactor and Other Critical Process Control Applications." In *FTCS*, pp. 462-469. 1991.

[6]. Piater, A., T. B. Ionescu, and W. Scheuermann. "A distributed simulation framework for mission critical systems in nuclear engineering and radiological protection." *International Journal of Computers, Communications & Control* 3, no. 3 (2008).