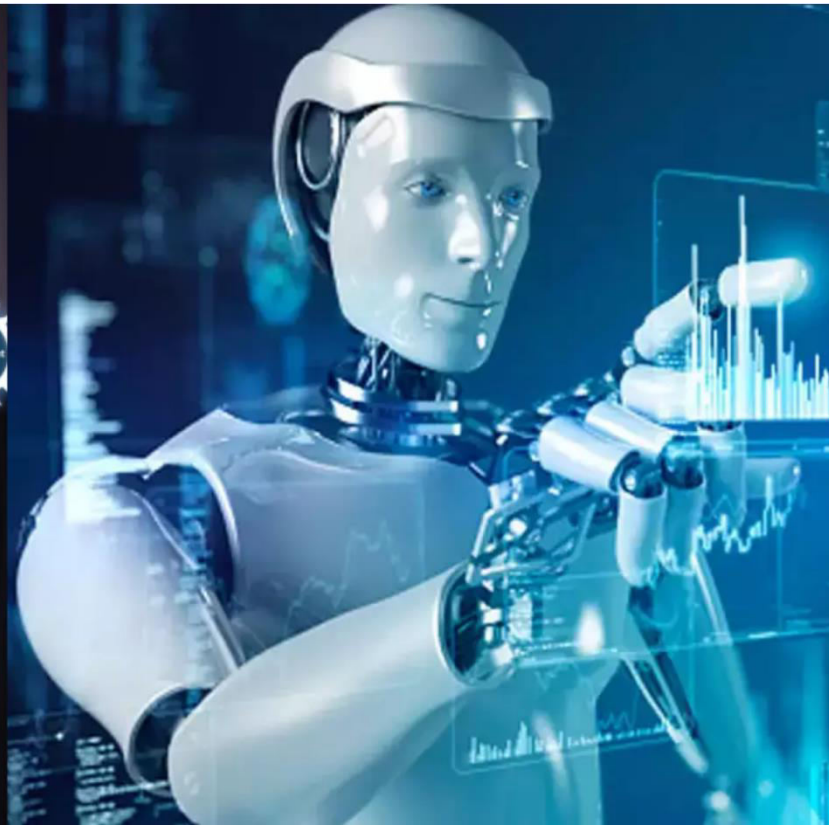
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

Real-Time Systems in Software Engineering: An In-Depth Perspective

Dr C K Gomathy, Mr. Phanindra Puneeth

Department of Computer Science and Engineering, SCSVMV University, Kancheepuram, Tamil Nadu, India

ABSTRACT: Real-time systems play a critical role in modern software engineering, where timely and predictable responses are essential for applications ranging from aerospace and automotive systems to industrial automation and healthcare. This paper provides an in-depth perspective on real-time systems, exploring their fundamental principles, design challenges, and implementation strategies. Key topics include the classification of real-time systems (hard, soft, and firm), scheduling algorithms, resource management, and the integration of real-time systems with emerging technologies such as the Internet of Things (IoT) and artificial intelligence (AI). The paper also examines the trade-offs between performance, reliability, and safety, highlighting best practices for developing robust real-time systems. By addressing both theoretical and practical aspects, this study aims to offer valuable insights for researchers and practitioners in the field of software engineering.

KEYWORDS: Real-Time Systems, Software Engineering, Hard Real-Time, Soft Real-Time, Scheduling Algorithms, Resource Management, IoT, Artificial Intelligence, System Reliability, Safety-Critical Systems.

I. INTRODUCTION

In the realm of software engineering, real-time systems stand out as a critical area of focus, marrying the intricacies of software design with the demands of time-sensitive applications. At its core, a real-time system is defined as one that must respond to inputs or events within a specific timeframe to ensure correct functionality. This characteristic distinguishes real-time systems from conventional software, where the timing of responses is less critical. The importance of real-time systems cannot be overstated, especially in fields such as telecommunications, automotive, aerospace, and healthcare. These systems are designed to handle tasks that must be executed precisely and predictably, where delays could lead to catastrophic failures or compromised safety. For instance, in an aircraft's navigation system, even a minor delay in processing data can result in severe consequences, highlighting the necessity for robust real-time capabilities.

This article aims to delve into the intricate world of real-time systems, exploring their foundational principles, key characteristics, and the unique challenges they present in software engineering. We will discuss the various types of real-time systems—hard, firm, and soft—and examine how they are applied in different industries. Additionally, we will consider the methodologies and tools that facilitate the development of these systems, along with the evolving landscape of real-time computing in our increasingly interconnected world. By the end of this exploration, we hope to illuminate the vital role that real-time systems play in ensuring not only operational efficiency but also safety and reliability in critical applications.

II. TYPES OF REAL-TIME SYSTEMS

Real-time systems are designed to perform tasks within a specific time frame to ensure reliability and efficiency. Each type serves unique needs and applications, balancing timeliness and resource usage based on the criticality of the operations involved.

A. Hard Real-Time Systems

1. **Definition and Characteristics**
   Hard real-time systems are those in which completing tasks on time is crucial. Missing a deadline in such systems can lead to severe consequences, from equipment damage to life-threatening situations. In these systems, every operation is planned down to the microsecond to ensure that tasks are completed within a

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

strictly defined timeframe. These systems typically require highly optimized hardware and software, and they demand extensive testing and reliability checks to function as intended.

Characteristics:

- Deterministic execution, with strict time constraints.
- High reliability and predictability in task completion.
- Often use specialized hardware with real-time operating systems (RTOS).
- Performance and timing are prioritized over other non-essential tasks.

2. **Examples and Applications**

Hard real-time systems are commonly found in applications where timing is paramount:

- **Automotive systems:**Airbag deployment systems, anti-lock braking systems, and electronic control units (ECUs) in cars.
- **Medical devices:**Heart rate monitors, insulin pumps, and defibrillators rely on precise timing to ensure patient safety.
- **Aerospace and defense:**Control systems in fighter jets, missile guidance, and satellite systems where timing is critical to operation.

B. Soft Real-Time Systems

1. **Definition and Characteristics**

Soft real-time systems are more flexible in terms of timing. While they still aim to meet deadlines, occasional delays or timing failures do not result in critical failures. These systems focus on providing a high-quality user experience or ensuring the best possible timing but allow for minor fluctuations without severe consequences. Soft real-time systems are common in consumer devices where smooth operation is essential but not life-threatening.

Characteristics:

- Timing is important, but occasional delays are acceptable.
- Prioritizes overall user experience over strict timing adherence.
- Can run on general-purpose operating systems with real-time adjustments.
- Usually emphasizes high availability and resource utilization without strict guarantees.

2. **Examples and Applications**

Soft real-time systems appear in various applications:

- **Multimedia streaming:**Video conferencing and streaming services, where delays are noticeable but not catastrophic.
- **Telecommunications:**Systems that manage call handling and routing aim for low latency, but minor delays are usually tolerable.
- **Online gaming:**Multiplayer games require real-time responses but can tolerate brief lags without breaking the game experience entirely.

C. Firm Real-Time Systems

1. **Definition and Characteristics**

Firm real-time systems sit between hard and soft real-time systems in terms of strictness. Missing a deadline in a firm real-time system may not be as critical as in a hard real-time system, but it still results in some degradation in quality or output. For example, if a firm real-time system misses a task deadline, the output from that task is often considered useless, and there may be a drop in service quality. However, unlike hard real-time systems, occasional missed deadlines do not lead to catastrophic outcomes.

Characteristics:

- Deadlines are essential, but occasional misses are tolerated without severe effects.
- Typically, missed deadlines lead to the discarding of the late output.
- More tolerant than hard real-time systems but less flexible than soft real-time systems.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

- Often used in systems where data or results are time-sensitive but can be bypassed or recalculated if a deadline is missed.

2. **Examples and Applications**
   Firm real-time systems are common in:
   - **Financial systems:**High-frequency trading platforms where processing delays can result in missed opportunities, though the system can recover in subsequent trades.
   - **Manufacturing automation:**Systems that handle non-critical processes but require timeliness for quality assurance, such as assembly line inspections.
   - **Inventory and logistics management:**Barcode scanners and robotic sorting systems that need to process information quickly but can tolerate occasional delays or retry mechanisms.

III. KEY CHARACTERISTICS OF REAL-TIME SYSTEMS

Real-time systems are defined by specific characteristics that ensure their reliable, timely, and predictable operation across various applications.

A. Timing Constraints

Timing constraints are central to real-time systems, as they dictate how quickly tasks need to be completed:

- **Deadlines:**Tasks must be completed within set deadlines to prevent failures. Hard real-time systems require strict adherence to deadlines, while soft real-time systems allow minor delays.
- **Response Times**: Real-time systems are designed to respond quickly to inputs, particularly in applications requiring immediate reactions, like medical or safety systems.
- **Periodic Tasks:**Many tasks in real-time systems are periodic, meaning they must execute at regular intervals (e.g., sensor data collection), which requires precise scheduling.

By effectively managing timing constraints, real-time systems maintain consistent responsiveness crucial for their applications.

B. Reliability and Stability

Reliability and stability are essential in ensuring that real-time systems operate without unexpected errors:

- **Reliability:**These systems must function dependably, as failures could disrupt essential processes or endanger lives. Reliability is achieved through rigorous testing and error detection.
- **Stability:**Real-time systems need to handle varying workloads without compromising performance, ensuring consistent operation under diverse conditions.

High reliability and stability make real-time systems suitable for critical applications, where unexpected downtimes are unacceptable.

C. Predictability

Predictability in real-time systems ensures consistent and repeatable behavior, enabling users to rely on the system's timing and performance:

- **Task Scheduling:**Predictable scheduling ensures tasks run in a known order within expected timeframes, often achieved with priority-based scheduling.
- **Deterministic Behavior:**Determinism enables the system to produce the same outcome in response to given inputs, ensuring timing and performance stay consistent.

Predictable operations are essential in real-time systems, providing consistency in timing and results, especially in time-sensitive applications.

D. Resource Management

Efficient resource management allows real-time systems to meet timing requirements without straining their capabilities:

- **CPU and Memory Allocation:**Managing CPU and memory resources ensures that high-priority tasks are completed on time.
- **I/O Management:**I/O devices are managed to prevent bottlenecks, enabling smooth data processing and timely responses.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

- **Power Efficiency:**Many real-time systems, particularly embedded devices, balance timing constraints with power conservation to extend battery life.

Resource management ensures real-time systems operate within hardware limits, maintaining performance and meeting deadlines.

IV. SOFTWARE ENGINEERING PRINCIPLES FOR REAL-TIME SYSTEMS

A. Requirements Engineering

1. **Gathering and Analyzing Requirements**
   Real-time systems require a clear understanding of functional needs and timing requirements. This involves gathering input from stakeholders and identifying scenarios that demand precise timing. Effective analysis helps pinpoint potential timing issues early in the process.

2. **Specifying Timing Constraints**
   Timing constraints, whether hard or soft, dictate the system's response time requirements. Documenting these constraints with timing diagrams and scheduling models guides the development process, ensuring that the system meets critical timing demands.

B. Design Methodologies

1. **Architecture Considerations**
   Real-time architectures often isolate time-critical components to ensure predictable performance. Layered or component-based designs help manage timing dependencies and resource allocation, balancing responsiveness with reliability.

2. **Design Patterns for Real-Time Systems**
   Patterns like the Observer (for updates), Scheduler (for task prioritization), and State (for managing operating states) are particularly useful in real-time systems. These patterns streamline development, address common timing challenges, and enhance system reliability.

C. Implementation Strategies

1. **Language and Tooling Considerations**
   Languages like C, C++, and Ada, along with RTOS (e.g., FreeRTOS, QNX), offer the low-level control needed for real-time systems. These tools support precise timing analysis and debugging, helping developers meet strict performance needs.

2. **Code Optimization Techniques**
   Optimization techniques—such as loop unrolling, minimizing context switches, and using fixed-point arithmetic—help reduce delays and improve timing accuracy. Profiling and timing analysis tools aid in identifying and refining critical sections of code.

V. REAL-TIME OPERATING SYSTEMS (RTOS)

Real-Time Operating Systems (RTOS)are specialized systems designed to meet strict timing requirements for real-time applications. Unlike general-purpose OS, RTOS focuses on predictability and reliability in task execution, which is crucial for applications in fields like automotive, aerospace, and industrial automation.

A. Definition and Purpose

An RTOS is built to execute tasks within specific time limits. Its main purpose is to ensure predictability and precision, enabling reliable performance even in high-stakes environments. By managing resources and prioritizing tasks, an RTOS allows critical operations to run on time, making it essential in real-time applications.

B. Key Features of RTOS

1. **Task Scheduling**
   Task scheduling is essential in an RTOS. Using priority-based scheduling, high-priority tasks preempt lower-priority ones, ensuring that critical operations meet their deadlines.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

- **Preemptive Scheduling:**Allows high-priority tasks to interrupt lower-priority ones.
- **Deterministic Scheduling:**Guarantees predictable execution within set timeframes.
- **Multitasking:**Manages multiple tasks efficiently for real-time performance.

2. **Inter-Process Communication (IPC)**
   RTOS must allow tasks to communicate quickly and without conflict. IPC mechanisms, like message queues and semaphores, help maintain real-time efficiency.
   - **Message Queues:**Organize task data exchange with prioritization.
   - **Semaphores and Mutexes:**Prevent conflicts over shared resources.
   - **Events and Signals:**Trigger immediate task responses to important changes.

C. Popular RTOS Examples

1. **FreeRTOS**
   FreeRTOS is an open-source RTOS known for its small footprint, making it ideal for IoT and embedded applications. It is portable, flexible, and well-documented.

2. **VxWorks**
   A commercial RTOS widely used in critical systems like aerospace, VxWorks offers advanced safety features and certifications. It's popular for applications requiring high reliability and security.

3. **QNX**
   QNX, with its microkernel design, excels in stability and security. Common in automotive and medical devices, it allows for modularity and real-time responsiveness.

VI. CHALLENGES IN REAL-TIME SYSTEM DEVELOPMENT

Real-time systems present unique challenges due to their critical timing requirements, resource constraints, and high reliability needs.

A. Complexity and Scalability

Real-time systems grow complex as they scale, requiring precise timing across multiple interconnected components.

- **Dependency Management**: Interdependent components increase the risk of cascading delays.
- **Scalability**: Expanding functionality without disrupting timing is challenging.
- **Scheduling**: Advanced algorithms are needed to manage tasks and ensure deadlines.

B. Managing Resource Constraints

Real-time systems often operate with limited processing power, memory, or energy, especially in embedded systems.

- **Memory and CPU**: Efficient resource management is crucial for preventing bottlenecks.
- **Energy Efficiency**: Battery-powered devices, like sensors, must be optimized for energy conservation.
- **Performance vs. Resources**: Developers must balance timing needs with resource limitations.

C. Debugging and Testing

Debugging real-time systems is complex due to their strict timing constraints.

- **Timing-Related Bugs**: Small timing deviations can cause failures, making bugs harder to detect and fix.
- **Reproducibility**: Timing issues may appear inconsistently, complicating testing.
- **Simulation Limitations**: Simulations may not fully reflect real-world conditions, making testing less predictable.

D. Evolving Requirements

Real-time systems often need to adapt to new regulations, technologies, or requirements, which can affect system stability.

- **Adaptability vs. Stability**: Changes must be implemented carefully to avoid disrupting critical functionalities.
- **Compliance**: Systems in fields like healthcare and automotive must meet strict regulatory standards.
- **Compatibility**: Updates must be compatible with existing hardware to maintain performance.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

VII. CASE STUDIES

Examining real-world applications of real-time systems across different industries illustrates their essential role in ensuring safety, efficiency, and user experience. Here's a look at some key examples:

A. Example 1: Real-Time Systems in Automotive Applications

In the automotive industry, real-time systems are integral to safety and performance. Modern vehicles are equipped with numerous control units, sensors, and microprocessors that work together to provide a seamless and secure driving experience.

- **Anti-lock Braking System (ABS):**ABS prevents wheels from locking during sudden braking, allowing the driver to maintain control over steering. This system must react within milliseconds to sensor data to modulate braking pressure, ensuring optimal stopping power without skidding.
- **Airbag Deployment Systems:**Airbags must deploy within milliseconds of detecting a collision to protect passengers. This requires a hard real-time system that can instantly interpret sensor data and trigger deployment without delay.
- **Electronic Stability Control (ESC):**ESC helps drivers maintain control of the vehicle during skids by applying brakes to individual wheels as needed. This system continuously monitors the vehicle's speed, wheel angle, and yaw rate, adjusting rapidly in real time to prevent loss of control.

Significance:

These real-time systems have drastically improved vehicle safety, reducing accident rates and saving countless lives. The precise timing and rapid response are what make these systems crucial in situations where even a split-second delay could have severe consequences.

B. Example 2: Real-Time Systems in Telecommunications

The telecommunications industry depends on real-time systems to manage vast volumes of data and ensure reliable connections across global networks. These systems are designed to handle large-scale data transmission and prioritize tasks based on timing and quality-of-service requirements.

- **Voice Over IP (VoIP) Services:**VoIP, like Zoom or Skype, relies on soft real-time systems to transmit voice and video data with minimal delay. While brief delays are tolerable, consistent timing is necessary for clear, uninterrupted calls.
- **Cellular Network Handover:**When a user moves from one cell tower's coverage area to another, the system must transfer the connection seamlessly. This real-time process ensures continuous connectivity without dropping calls.
- **Network Traffic Management:**Telecom systems use real-time traffic management to prioritize data packets, balancing between high-priority tasks (like emergency calls) and general data (such as social media). This helps manage bandwidth efficiently and improves user experience.

Significance:

Real-time systems in telecommunications allow billions of people to stay connected reliably. They ensure the smooth flow of information, even under high network load, and provide the backbone for critical applications such as emergency communication and remote work.

C. Example 3: Real-Time Systems in Healthcare

In healthcare, real-time systems are essential for monitoring, diagnostics, and life-saving treatments. From tracking patient vitals to guiding surgical robots, these systems play a critical role in improving medical outcomes and patient safety.

- **Patient Monitoring Systems:**In intensive care units, real-time monitoring systems track patients' vital signs, such as heart rate, blood pressure, and oxygen levels. Any significant fluctuation triggers an alert for immediate response, making these systems indispensable in critical care.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

- **Medical Imaging and Diagnostic Tools:**MRI and CT scanners rely on real-time data processing to create clear, accurate images. Real-time data helps radiologists identify issues quickly and provides precise imaging for diagnosis and treatment planning.
- **Robotic Surgery:**Robotic surgical systems enhance precision in complex procedures by providing surgeons with real-time feedback and control. These systems integrate sensor data to monitor patient responses, enabling immediate adjustments to surgical actions.

Significance:
Real-time systems in healthcare have transformed patient care, enabling faster diagnoses, more accurate monitoring, and safer surgical procedures. Their ability to provide immediate responses to changes in a patient's condition can be lifesaving and has set new standards in medical care.

VIII. FUTURE TRENDS AND DIRECTIONS

The landscape of real-time systems is evolving rapidly, driven by technological advancements that enhance performance and expand capabilities.

A. Advances in Real-Time Computing

Real-time computing is experiencing significant advancements due to improvements in hardware and software. High-performance computing (HPC) solutions are becoming more accessible, enabling complex computations to be processed in real-time. Furthermore, modern real-time operating systems (RTOS) are optimizing multi-core processors for better performance and reliability.

Virtualization is also gaining traction, allowing developers to create isolated environments for different tasks. This enhances scalability and simplifies system maintenance while improving resource allocation and stability.

B. Integration with IOT and Edge Computing

The integration of real-time systems with the Internet of Things (IoT) and edge computing is reshaping data processing. By analyzing data closer to the source, these systems reduce latency and bandwidth consumption.

For example, in smart cities, real-time systems can optimize traffic signals based on data from sensors and cameras. In healthcare, wearables can monitor patients' vital signs instantly, enabling immediate responses to critical changes.
Edge computing also enhances resilience, allowing critical operations to continue even during server outages, which leads to more responsive and adaptive systems.

C. Artificial Intelligence in Real-Time Systems

Artificial Intelligence (AI) is set to transform real-time systems by enabling predictive analytics and intelligent decision-making. For instance, AI can predict equipment failures in manufacturing by analyzing sensor data in real-time, reducing downtime.

In customer service, AI-driven chatbots can provide immediate, tailored responses based on real-time interactions. The fusion of AI with real-time systems will not only enhance responsiveness but also allow these systems to learn and adapt, leading to improved performance over time.

IX. CONCLUSION

A. Summary of Key Points

This article examined the different types of real-time systems, highlighting their definitions, characteristics, and applications. Hard real-time systems adhere strictly to deadlines, critical in fields like automotive safety and medical devices. In contrast, soft real-time systems allow for occasional delays, making them suitable for applications such as multimedia streaming and online gaming. Firm real-time systems occupy a middle ground, where missed deadlines result in discarding outputs rather than critical failures. Together, these classifications underscore the diverse requirements and roles of real-time systems in technology today.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

B. The Importance of Continued Research in Real-Time Systems

As technology evolves, so do the demands on real-time systems. Ongoing research is vital to enhance reliability, efficiency, and performance, particularly in areas like the Internet of Things (IoT) and autonomous vehicles. Innovations in algorithms and hardware design can improve the capabilities of these systems, ensuring they meet the growing complexities of modern applications.

C. Final Thoughts

Real-time systems are crucial in our interconnected world, driving advancements across multiple sectors. Understanding the differences between hard, soft, and firm real-time systems helps engineers design solutions that ensure safety and efficiency. By committing to research and development in this field, we can pave the way for future innovations that enhance performance in systems requiring timely and precise operations.

REFERENCES

1. C.K.Gomathy.(2010),"Cloud Computing: Business Management for Effective Service Oriented Architecture" International Journal of Power Control Signal and Computation (IJPCSC), Volume 1, Issue IV, Oct - Dec 2010, P.No:22-27, ISSN: 0976-268X .
2. C.K.Gomathy and Dr.S.Rajalakshmi.(2011), "Business Process Development In Service Oriented Architecture", International Journal of Research in Computer Application and Management (IJRCM) ,Volume 1,Issue IV, August 2011,P.No:50-53,ISSN : 2231-1009.
3. C.K.Gomathy and Dr.S.Rajalakshmi.(2012),"An Efficient Business Integration and Quality Service for Service Oriented Architecture", World Academy of informatics and management sciences(WAIMS), Volume 1,Issue II (WAIMS/0019/0084), March 2012,P.No: 74-79, ISSN : 2278-1315.
4. C.K.Gomathy and Dr.S.Rajalakshmi.(2012),"A proficient Business incorporation and Quality for SOA. ", International journal on information Science and computing, Volume 6,Issue II,July-2012, ISSN : 0973-9092.
5. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"A Business Intelligence Network Design for Service Oriented Architecture", International Journal of Engineering Trends and Technology (IJETT) ,Volume IX, Issue III, March 2014, P.No:151-154, ISSN:2231-5381.
6. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"Software Architecture Design Using Service Oriented on Quality Metrics", Australian Journal of Computer Science (AUJCS), Volume I,Issue I March 2014,P.No:09-16,ISSN:2251-3221.
7. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"Software Pattern Quality Comportment In Service Oriented Architectures", European Scientific Journal (ESJ) volume-10,No-9,Issue-March 2014,P.No-412-423,ISSN-1857-7881.
8. C.K.Gomathy and Dr.S.Rajalakshmi.(2014), "A Software Design Pattern for Bank Service Oriented Architecture", International Journal of Advanced Research in Computer Engineering and Technology(IJARCET), Volume 3,Issue IV, April 2014,P.No:1302-1306, ,ISSN:2278-1323.
9. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"A Software Ability link for Service Oriented Architecture", Global Journal of Management and Business Research (GJMBR), Volume(A) XIV, Issue-II, Version 1.0,May 2014, P.No:11-14, ISSN:2249-4588.
10. N.Sugumar, C.K.Gomathy, ”Smart LCM for energy-efficient mechanism using CTX in wireless sensor networks” International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume No.1 ,issue No. 4,IISSN-2348-4853.P.No 47-56
11. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"A Software Ability Network in Service Oriented Architecture", International Journal of science and Technology Education Research(IJSTER) , Volume 5,Issue II, June 2014,P.No:7-14, ISSN:2141-6559.
12. C.K.Gomathy and Dr.S.Rajalakshmi.(2014),"A Software Quality Metric Performance of Professional Management in Service Oriented Architecture", Proceedings of ICCTET’14, organized by Akshaya College of Engineering, Coimbatore. Archived in IEEE Xplore Digital Library, July 2014,ISBN:978-1-4799-7986-8.
13. C.K.Gomathy and Dr.S.Rajalakshmi.(2014), "Service Oriented Architecture to improve Quality of Software System in Public Sector Organization with Improved Progress Ability", Proceedings of ERCICA-2014, organized by Nitte Meenakshi Institute of Technology, Bangalore. Archived in Elsevier Xplore Digital Library, August 2014, ISBN:978-9-3510-7216-4.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403117|

14. C.K.Gomathy Dr.S.Rajalakshmi, M.Prema,"A framework for developing service-oriented architecture for mobile commuting-an exploratory study of SCSVMV university "Journal of Harmonized research in Engineering (JOHR) Volume No.2, Issue No.IV, October 2014,P.No:360-366, ,ISSN : 2347-7393
15. N.Sugumar, C.K.Gomathy," Energy-efficient mechanism using CTX In wireless sensor networks" Zenith International Journal of Multidisciplinary Research ,ISSN 2231-5780 Volume No.4 (5), MAY (2014), pp. 176-188.
16. C K Gomathy and V Geetha. Article: A Real Time Analysis of Service based using Mobile Phone Controlled Vehicle using DTMF for Accident Prevention. International Journal of Computer Applications 138(2):11-13, March 2016. Published by Foundation of Computer Science (FCS), NY, USA,ISSN No: 0975-8887
17. Dr.C K Gomathy and Dr.V.Geetha,Fake Job Forecast Using Data Mining Techniques, International Journal of Early Childhood Special Education (INT-JECSE)
18. Dr.C K Gomathy and Dr.V.Geetha, Multi-Source Medical Data Integration AndMining For Healthcare Services, International Journal of Early Childhood Special Education (INT-JECSE) DOI: DOI:10.9756/INTJECSE/V14I5.67ISSN: 1308-5581 Vol 14, Issue 05 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details