Volume : 1 | Issue : 12 | September 2012 ISSN - 2249-555X

Research Paper Computer Science

Application of Computer & Engineerning: Applying Real Time System

*Chandrakant D. Prajapati **Sanjay B. Patel

*,** Asst. Prof., Department of Computer Science, Ganpat University, Kherva, Mehsana, Gujarat, India

ABSTRACT

The paper briefly presents the computer technologies applied in computer science implemented by the Engineering, including a graphical user interface, database, real-time, multimedia and networking, etc. computer application and technologies is highly independent and rapidly advancing technologies: object-oriented methods and real-time systems. It can be used for systematically developing the software of embedded real-time systems gaining the benefits of the object-orientation as commonly reported for the development of other branches of software, but matching also the specific requirements of real-time systems. A real-time operating system (RTOS) supports real-time applications and embedded systems. Real-time applications have the requirement to meet task deadlines in addition to the logical correctness of the results.

A real-time operating system (RTOS) supports real-time applications and embedded systems. Real-time applications have the requirement to meet task deadlines in addition to the logical correctness of the results.

Keywords : Hard Real-Time, Soft Real-Time, Controlled system, Sensor, RTOS etc.

1. Introduction

Computer science or computing science is the scientific and mathematical approach to computation and specifically to the design of computing machines and processes. A computer scientist is a scientist who specializes in the theory of computation and the design of computers.

Its subfields can be divided into practical techniques for its implementation and application in computer systems and purely theoretical areas. Some, such as computational complexity theory, which studies fundamental properties of computational problems, are highly abstract, while others, such as computer graphics, emphasize real-world applications. Still others focus on the challenges in implementing computations. For example, programming language theory studies approaches to description of computations, while the study of computer programming itself investigates various aspects of the use of programming languages and complex systems and human-computer interaction focuses on the challenges in making computers and computations useful, usable, and universally accessible to humans.

Real-time systems have been defined as: “those systems in which the correctness of the system depends not only on the logical result of the computation, but also on the time at which the results are produced”. [1]

Real-Time systems are becoming universal. Typical examples of real-time systems include Air Traffic Control Systems, Networked Multimedia Systems and Command Control Systems etc. In a Real-Time System the correctness of the system behavior depends not only on the logical results of the computations, but also on the physical instant at which these results are produced. Real-Time systems are classified from a number of viewpoints i.e. on factors outside the computer system and factors inside the computer system. [2]

Real-Time systems span several domains of computer science. They are defense and space systems, networked multimedia systems, embedded automative electronics etc. In a real-time system the correctness of the system behavior depends not only the logical results of the computations, but also on the physical instant at which these results are produced. A real-time system changes its state as a function of physical time, e.g., a chemical reaction continues to change its state even after its controlling computer system has stopped. Based on this a real-time system can be decomposed into a set of subsystems i.e., the controlled object, the real-time computer system and the human operator. A real-time computer system must react to stimuli from the controlled object (or the operator) within time intervals dictated by its environment. [2]

The example of real time systems are Commands and Control systems, Air traffic control systems are examples for hard real-time systems. On-line transaction systems, airline reservation systems are soft real-time systems.

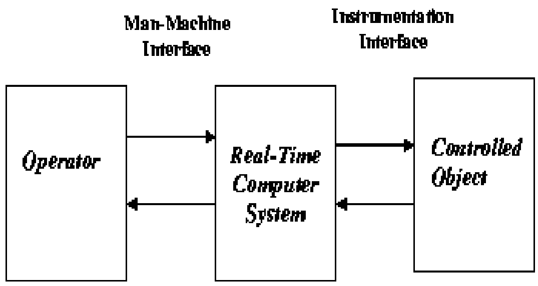

Figure 1: Real-Time System

2. Concepts of real time system

Real-Time systems can be confidential from different viewpoints. The first two classifications, hard real-time versus soft real-time, and fail-safe versus fail-operational, depend on the characteristics of the application, i.e., on factors outside the computer system. The second three classifications, guaranteed-timeliness versus best-effort, resource-adequate versus resource-inadequate, and event-triggered versus time-triggered, depend on the design and implementation, i.e., on factors inside the computer system.

2.1 Hard Real-Time versus Soft Real-Time

The major differences between hard and soft real-time systems are as follow. The response time requirements of hard real-time systems are in the order of milliseconds or less and can result in a tragedy if not met. In contrast, the response time requirements of soft real-time systems are higher and not very stringent. In a hard real-time system, the peak-load performance must be predictable and should not violate the predefined deadlines. In a soft real-time system, a degraded operation in a rarely occurring peak load can be tolerated. A hard real-time system must remain synchronous with the state of the environment in all cases. On the other hand soft real-time systems will slow down their response time if the load is very high. Hard real-time systems are often safety critical. Hard real-time systems have small data files and real-time databases. Temporal accuracy is often the concern here. Soft real-time systems for example, on-line reservation systems have larger databases and require long-term integrity of real-time systems. If an error occurs in a soft real-time system, the computation is rolled back to a previously established checkpoint to initiate a recovery action. In hard real-time systems, roll-back/recovery is of limited use. [3]

For hard real-time systems, all tasks should not miss their deadline otherwise a big problem can Happen. One example of such systems is nuclear reactors. Soft real-time systems accept some Misses in deadline, for example in mobile phone where missing some deadlines will decrease the quality of the sound. Many systems are mixture of hard and soft tasks.

3. Application of real time system

Real-Time Operating Systems (RTOS) can be used to provide predictable services to the applications. RTOS provide the primitives real-time scheduling policies; inter process communication and run-time monitoring. There a number of RTOSs, e.g. RT-Mach, Solaris, Lynx.

Real-Time systems interact with their environment by input/output subsystem. Sensors and actuators are the examples of i/o elements in real-time systems. On the other hand i/o an important part of real-time systems.

Fault tolerance is important in safety-critical real-time systems because otherwise a single component failure can lead to a catastrophic systems failure.

With the growth of Internet several multimedia applications like multimedia are merging with real-time systems. Scheduling in these systems is done to provide good quality of service. Some of the real-time systems research is being extended to QoS scheduling to multimedia applications.

4. Working on real time system.

A control system communicating with sensors and actuators over a communication network will be called a distributed real-time control system. In distributed real-time control systems, figure shows

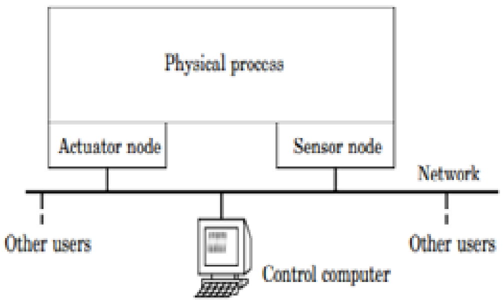

Figure 2. Distributed real-time control system with sensor node, controller node and actuator node.

Data are sent and received by network nodes of different kind and manufacturers. Network nodes that are of specific interest for distributed control are sensor nodes, actuator nodes, and controller nodes. Sensor nodes measure process values and transmit these over the communication network. Actuator nodes receive new values for the process inputs over the communication network and apply these on the process input. Controller

nodes read process values from sensor nodes. Using a control algorithm control signals are calculated and sent to the actuator nodes. The system setup with a common communication network reduces cost of cabling and offers modularity and flexibility in system design

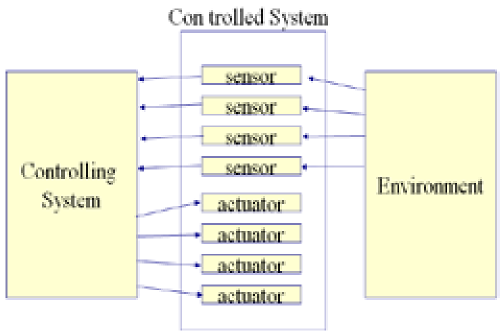

Figure 3. Controlled system

Real-time systems often are comprised of a controlling system, controlled system and environment.

Controlling system: acquires information about environment using sensors and controls the environment with actuators.

Timing constraints: derived from physical impact of controlling systems activities. Hard and soft constraints.

4. Conclusions

Real-Time systems span a large part of computer industry. So far most of the real-time systems research has been mostly confined to single node systems and mainly for processor scheduling. This needs to be extended for multiple resources and distributed nodes. Real-time systems are expanding to several other domains such as automative industry and embedded real-time systems. Especially the marriage of the Internet with multimedia applications has opened several new volume applications. Most of real-time control applications are developed in two stages; control design stage and computer implementation stage. In the control design stage, the control engineers use design theories to evaluate control model that make the overall system fulfill the desired performance. In the computer implementation stage, the designed control model is converted into hard real-time periodic tasks called control tasks.

REFERENCES

[1] J. Stankovic, "Misconceptions About Real-Time Computing," IEEE Computer, 21(10), October 1988. | [2]http://www.ece.cmu.edu/~koopman/des_s99/real_time/ | [3] [Kopetz97] Kopetz, H., Real-Time Systems, Design Principles for Distributed Embedded Applications, Klower Academic Publishers, 1997 | [4] NILSSON, J H1996I: "Analysis and design of real-time systems with random delays." Report ISRN LUTFD2/TFRT- -3215- -SE. | [5] Kang G, Xianzhong C. "Computing Time Delay and Its Effect on Real-Time Control Systems". IEEE transaction on control systems tech. Vol. 3, NO 2, June 1995. | [6] Kopetz, H., Real-Time Systems, Design Principles for Distributed Embedded Applications, Klower Academic Publishers, 1997, Chpt. 10-11.