Intelligent Real-Time Systems for Managing Catastrophe through Scenario Shift Paradigm

T.R. Gopalakrishnan Nair
Director, Advanced research Centre,
Realtime Complex Systems Division,
RRG Institutions, Bangalore, India.

A. Christy Persya
Associate Professor, BNMIT,
VTU, Bangalore, India.

ABSTRACT

The intelligent real-time system design needs to incorporate autonomic features in their operations to achieve the unexpected criticalities of systems and its environment. Catastrophic scenarios can emerge in systems, challenging the traditional role of real-time systems where the temporal rigidity is the essential design feature. The priorities and its management scheme given for a normal operation by the conventional real-time systems need not be the ultimate format to meet the requirements of a catastrophic environment. Hence, usual real-time system is supplemented with a layer of intelligence to deal with the emerging catastrophic environment. Intelligent real-time systems can have hybrid schedulers with some additional features that can guarantee risk mitigation performance even with the occurrence of extreme, unusual variations of external conditions. This approach addresses intelligence in systems by making a real-time system schedule itself to adapt meaningfully even if the environment changes, by assigning intelligent priorities. This paper introduces the design of Intelligent Real-Time System (IRTS) that keep shifting the boundaries of the original hybrid scheduler with cognitive features aiding the intelligence by increasing the possibility to make a dynamically reconfigured system while increasing the fairness of the scheduling. Intelligent scheduler can be used in embedded critical systems in order to cope with the unexpected problems like nuclear power plants and hazardous installations. Theoretical analysis shows that the proposed design performs the operation of IRTS, which can be advantageously applied to pragmatic systems and show how intelligence works with priority.

Keywords

Real-time system, catastrophe, intelligent, cognitive, priority

1. INTRODUCTION

The design of disaster management is one of the major issues in many real-time system applications in the recent past [14]. It may not be good enough if the systems that are expected to govern this catastrophe tend to work the way in which earlier real-time system used to respond. It calls for improved approaches in the design and implementation of governing systems. This paper deals with intelligence incorporated mainly in scheduling and demonstrates how it can play a vital role in unexpected scenarios. Usually, all real-time systems are designed and created with fixed schedulers and fixed patterns of performance under expected paradigms. However, systems can move from the initial scenario to a particular final scenario whereby the benign environment in which system was working changes to a hostile environment. In such cases, cognitive capabilities can be utilized by the system to face the situation meaningfully. For this purpose, systems need to be more intelligent in the real-time world which calls for Intelligent Real-Time System (IRTS) as shown in Fig 1.

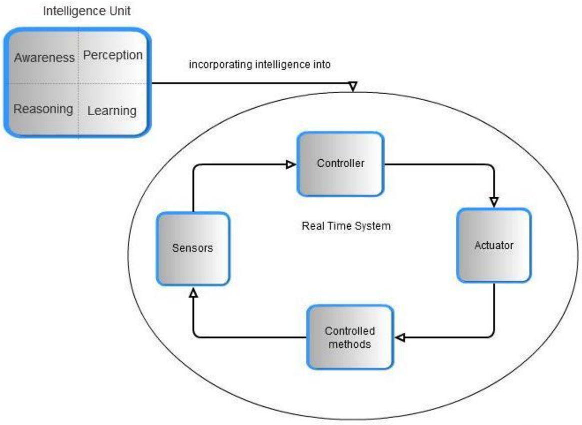

Fig 1. Incorporating Intelligence into RTS

There are many significant results achieved in the field of adaptive schedulers [4, 8]. In [3], Liang-Teh Lee et al have the techniques of feedback scheduling by triggering alert signals to improve the schedulers or by calculating the urgency of tasks and using them. However, a scheduler meant for catastrophe management has more responsibility in incorporating intelligence in the real-time scheduling scenario.

The major problem in manufacturing system is the dynamically changing system states which are the small and large unexpected deviations [15]. Here, rescheduling is the process of altering an existing schedule because of the disruptions caused or other changes. Based on rescheduling factors, methods like right shift scheduling, partial rescheduling and complete regeneration are identified.

The real-time system must be intelligent enough to monitor or observe and respond quickly by rescheduling the unexpected system conditions. Intelligent systems interact with schedulers, sequencers, scenario observers and safety monitors [5]. In a well-defined system, intelligence will hold the knowledge of different scenarios and further it can learn and improve the dealing capability. It can also detect scenario changes and try to implement new strategies based on environmental changes. Then it will reconfigure and upgrade itself. This upgradation results in state changes of all executing tasks at different levels. So, the logical decision organized suitable for nullifying the negative effects of the systems drifts disastrously or otherwise is called Intelligence.

The capability of being adaptive provides innovative interactions or solutions that can reduce the complexity of situations to bring down total severity. The real-time systems must be adaptive in the case of large-scale applications like nuclear power plants and large process plants to meet the environmental changes with the arrival of unusual critical tasks, to withstand the stability.

In view of this, this paper combines intelligence in the operations of scheduling to meet the unexpected criticalities of systems and its environment.

The paper is structured as follows: Section 2 illustrates the research background. In section 3, the concept of intelligence in real-time system is given, and fusion of intelligence with old real-time systems is designed in section 4. The operations of IRTS are provided in section 5. The conclusion is in section 6.

2. RESEARCH BACKGROUND

2.1 Scheduler Sequence

The current schedulers as shown in Fig 2 are listed as follows: Deterministic scheduling provides liable, objective, and consistent method to account for all the variables that could affect the desired outcome of the application.

Benefits are:

- Proven method for predicting system functionality
- Enables developers to streamline and accelerate the design process
- Lower development costs and improves business process flow
- Increases the quality and reliability
- Reduces the risk of missed product deadlines and failures
- Results in reliable, performance-tuned software that is more closely aligned with design goals

The commonly used real-time system scheduling scheme is fixed-priority pre-emptive scheduling. This scheme ensures that at any given time, the processor executes all the tasks which are ready to execute in a periodic manner. e.g.-Round Robin Scheduling.

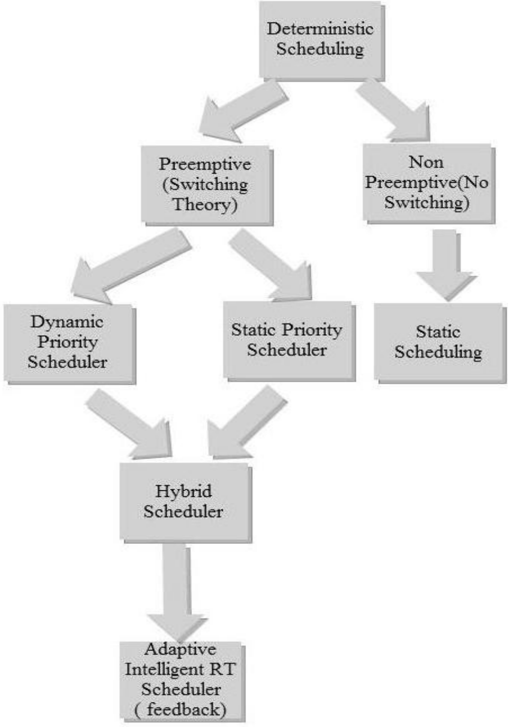
Deterministic Scheduling
Preemptive (Switching Theory)
Non Preemptive(No Switching)
Dynamic Priority Scheduler
Static Priority Scheduler
Static Scheduling
Hybrid Scheduler
Adaptive Intelligent RT Scheduler (feedback)

Fig.2 Types of Schedulers

The state in which the running process is not interrupted or halted by any other process until it finished its execution is called non-preemptive scheduling. Example includes first in first out Scheduling.

In dynamic priority scheduling, the priorities of tasks are calculated during the run time [9]. The goal of these scheduling is to adjust to dynamically changing situations. It is very hard to produce well-defined policies to meet the goals that depend on the difficulty of given problems [6]. Examples include earliest deadline first scheduling and least slack time scheduling.

One of the static priority scheduling scheme is Rate Monotonic scheduling algorithm which is used in many real-time operating systems [1]. The priorities are assigned statically based on the cycle duration of the job: the shorter the cycle period, the higher the jobs priority [2].

Hybrid Scheduling is a class of scheduling mechanism that mixes different scheduling criteria or disciplines in one algorithm. Adaptable schedulers allow the scheduling to change when the environment changes in a known way and adapt their behavior accordingly [10, 11].

Yun Niu and Guanzhony Dai have clearly explained the Reservation based feedback scheduling with simulation results [1]. The resource reservation can increase the performance and the effectiveness of the designed controller. They use the factor polynomial to put the poles to desired location, and construct a piecewise quadratic Lyapunov function to test the asymptotical stability of the algorithm [16].

Insop discussed the Feedback bandwidth server for scheduling [2]. The Feedback control based switched system with bounded time-varying uncertainty uses virtual finishing time of a dedicated server. There are advantages of using feedback control in scheduling. In such cases, the precise schedulability models are not required [7] because the feedback controller can adjust these parameters dynamically. Finally, they concluded by providing a systematic way for designing a feedback controller in terms of modeling, analyzing stability and synthesizing the controller.

2.1 Need for the next level of scheduling

The next level of real-time system is mixed criticality system. From 2007 to till date, various approaches in uniprocessor scheduling and multiprocessor scheduling are realized with realistic models and formal methods [17]. A mixed criticality system has two or more distinct levels like safety critical, mission critical and low critical [18]. In this model, the parameter worst case execution time is dependent on the criticality level. It means, the same parameter WCET can have a higher value for safety critical level and a different value for mission critical and low critical levels. The real-time applications work in a sequence of pre-defined modes. An automated train or any other system may have a start mode or initialize mode, cruise mode, and driving mode. The mode change protocols are required for the system to move between modes. These protocols are dealt and studied by various researchers in their study [18], [19], [20]. Mixed criticality system addresses on two levels such as high criticality level and low criticality level. These mode change protocols were dealt with three types of modes [21]. They are normal mode, exceptional mode and degraded mode. The proposed model is a continuous mode estimator which uses the four cognitive properties to learn the changes happening in the environment, and decides on an appropriate best fit scheduling mechanism to handle the critical situation.

3. INTELLIGENT REAL-TIME SYSTEM (IRTS)

Intelligent Scheduling, defined below, form the backbone of our new RTS. Recall that the priority (ꞵ) of a task T is the weight of the task with other tasks. Note that the priority of a task is always a positive integer.

Definition 3.1. An intelligent real-time system (IRTS) is a nature inspired, mathematically sound problem-solving tool that satisfies the following properties.

i. Awareness
ii. Perception
iii. Reasoning
iv. Learning

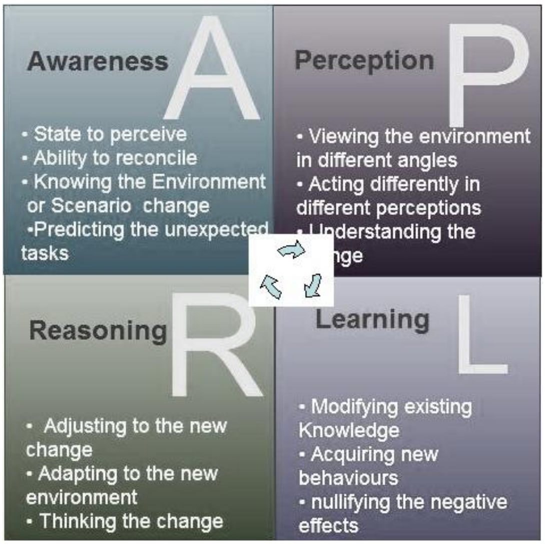
Awareness
A
• State to perceive
• Ability to reconcile
• Knowing the Environment or Scenario change
•Predicting the unexpected tasks
Perception
P
• Viewing the environment in different angles
• Acting differently in different perceptions
• Understanding the change
Reasoning
R
• Adjusting to the new change
• Adapting to the new environment
• Thinking the change
Learning
L
• Modifying existing Knowledge
• Acquiring new behaviours
• nullifying the negative effects

Fig 3. Cognitive Properties

As shown in Fig 3, awareness parameters are the parameters that are measurable and are useful to assess the adverse effects of the system. Perception is deep understanding or comprehension with at least a cause-effect model of the situation.

With a deep understanding, the system can reason its effect that is transformed into knowledge that is later usable, and this process is called Learning. The best example of real-time system is the human body. Human body has many real-time events running inside. When a lion attacks, the system does not work, in the same way, as it does. It changes by itself. Here, the system detects the scenario change and changes itself. Real-time system works, in the same way, with the regular schedule. The system itself will change its regular pattern of operation to counterbalance effects when catastrophic critical task enters and acts in super scheduler manner [12, 13]. The system will enable all possible resources like maximizing energy, communication, etc to manage the situation.

Learning will observe how much adverse change or effects are circulating in the system. The assessment can be on to assert what are the significant adverse effects of the system. The parameters that are required for learning will see the adverse effects. If the adverse effects are building because of certain other parameters, then the system starts working such a way that it throws the task which are actually already promoting negative effects and it will go down and make another set of parameters which will nullify the negative effects. Intelligence is required to learn and adjust these priorities. Priorities of previous no catastrophic scenario may produce adverse effects in the system. An excellent example is the post earthquake nuclear power plant scenario and change of priorities and modes of operation required by distributed controls and surveillance systems facing the challenges of a possible melting down of the core. Many of the usual real-time jobs of high priority become no more the highest priority, and new ones emerge with highest demands that are oriented towards containing catastrophe.

Lemma 3.1. *Priority value is inversely proportional to the quantity of the adverse effects of the system.*

$$P_v = \frac{K}{-S} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Where

K is the proportional constant
S is the System, and
P_v is the Priority Value

By incorporating priority-effect learning system, the adverse effects can be identified, characterized and can be controlled. So the manually assigned priority is not ultimate because if various other adverse effects are arriving, then the existing priority will go down, and a different decision system will be taking over.

Let NS(x) be an increasing function to determine the service provided to the normal RTS, CS(x) be an another increasing function to determine the service provided to catastrophe task when it enters the system with high severity and CSTY(x) be an decreasing function to determine the high severity level of the catastrophe entry. The parameter service and severity of tasks cut each other in a point, which is called as the mixed ratio catastrophic service performance.

In normal case, the real-time system has 4 main parameters such as release time, execution time, deadline and priority. When the intelligence unit is incorporated into the RTS, these four parameters will not be sufficient because they are codified by the system modeler of old real-time system practices. An additional item called intelligence layer is required to expand the RTS as shown in Fig 4.

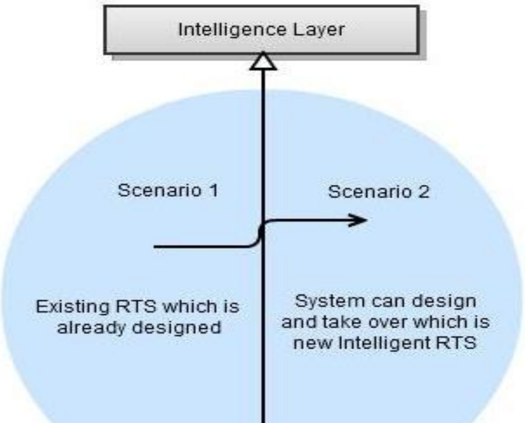
Intelligence Layer
Scenario 1
Scenario 2
Existing RTS which is already designed
System can design and take over which is new Intelligent RTS

Fig 4. Intelligence Layer in RTS

This explains that the existing RTS is modeled into Intelligent RTS with the strap of intelligence layer. The new RTS is modeled and designed with cognitive features of intelligence. Regularly RTS works in the existing way and changes itself when detects a scenario change and works in a different manner to avoid catastrophe situation.

The objectives of learning can be defined as

- Ability to understand the scenario shift.
- Ability to handle the mixing ratio of catastrophe and non catastrophic tasks.
- Ability to respond effectively to a new situation.

From the perspective of computation, intelligence of the system is characterized by its flexibility, adaptability and the ability to manage uncertain and imprecise information. Adaptive scheduling will be done in evaluating the complexity of real-time system. That intelligent layer is usually hidden in mind. Intelligent layer can meet the challenges raised in the embedded critical field.

4. FUSION OF OLD RTS WITH INTELLIGENCE UNIT

The old real-time systems have only the scheduler and the storage of the tasks while the new intelligent system incorporates the intelligent unit which drives the system from the initial level to the final level. Just like climbing a flight of stairs, the human intervention increases from initial to the final level. So, in the presence of unusual tasks or when there is environment change, the scheduler can act accordingly in that it reduces the catastrophic disaster. In few existing slack preservation schemes, when sporadic tasks arrive, based on their acceptance test a part of periodic tasks are rejected from the scheduled system in order to meet the deadlines of sporadic and aperiodic tasks. Here, when there is catastrophic change in the environment, the system stability is not disturbed based on any acceptance test. The new real-time system with its intelligence unit incorporated into the old real-time system is as shown in Fig 5.

It is natural that when unexpected incidents amounting to catastrophe evolves the level of the system cannot be kept as normal situations. If any increased demand originates, old schedulers will be working in their old scheme and try to shift itself best to the allowed boundary preconceived. But the scenario can be drastically different like Fukushima incident, Japan, 2011 or an Aircraft facing partial propulsion failures that is like an environmental changes from level 1 to level 2 to level N. The typical levels of building up catastrophe in each case will be under the learning process of intelligence.

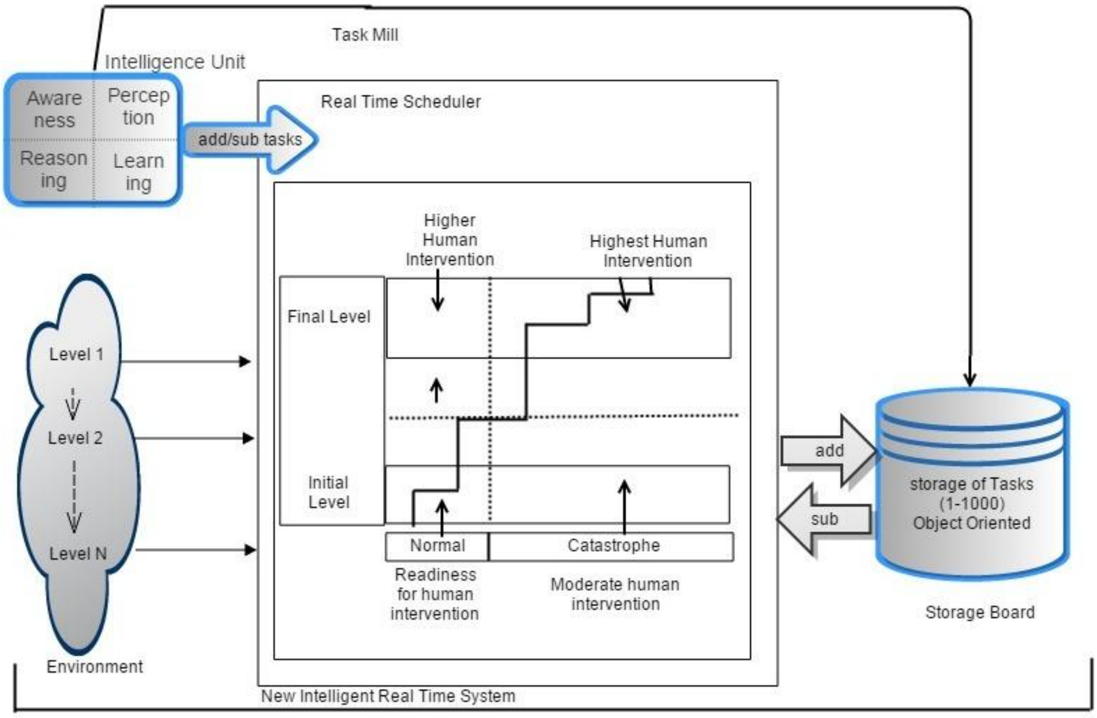

Fig 5. Intelligent Real-time System (IRTS)

According to the scenario assessment, intelligence will be added or subtracted task from storage to scheduler and shift the focus or bias of the whole system to very close extremes enabling catastrophe contingency forces. The number of tasks and their priority existing tasks will be changed depending on the addition or reduction based on the environment level change. Here the output of intelligence can be organized primarily for the scheduler reconfiguration and internal subsystem performance priority changes along with shifting internal system compensation. It calls for addition or subtraction of tasks. Intelligence unit must have knowledge and must have all most all tasks ready in the store. It must also have the capability to couple more than one task and create a combinational task which must be suitable to face current system bias changes to catastrophe. Such learning based task combinations will be synthesized, be picked up and released at any time.

Intelligent unit will be connected to the storage tanks functions and compensators, so the intelligence unit can know what the tasks available in the store are and properly orient the task mill which is the scheduler. The storage board has two parts i) tasks & ii) the object. So intelligence will look into the environment and the task storage board, when environment level changes. It can add or remove tasks to the task mill from storage board based on properties, awareness, and perception, reasoning and surely based on learning till now. Learning and action will create the new task array to develop the whole scenario management. When any real-time system is under threat then both the catastrophic task and non catastrophic task should be taken care, so a perfect autonomic time test will be done which will increase the cycle time. In any real time system exertions will not be allowed into the catastrophic window. This will be filled up only when the scenario changes, otherwise all the critical tasks will be executed and there will be a gap of no activity. There will be reserve dead band.

4.1 Operation of IRTS

The operations of IRTS are described by the steps given below. As a result of these operations, system stability is hoard to guarantee the quality of performance desired in each environment change. The description that intends to give an overview and leaves out details those are not essential for the understanding of IRTS method.

IRTS Operations

- Tasks are created and maintained for each environment change in storage board
  - The old scheduler schedules usual periodic, aperiodic and sporadic tasks.
  - The new reconfigured scheduler schedules unusual tasks which can arrive from environment level change along with tasks created through intelligence unit.
- Eenvironment level change brings the increase or decrease of tasks moved from task storage board into RTS
  - The pattern of addition or subtraction of tasks happens in such a way that in the case of a human being who walks swinging freely his hands and when it rains suddenly, he opens the umbrella and hold it in one hand. Here the opened umbrella is the additional subsystem called in, and only one hand cannot be swung anymore. That hand functions as a control of the umbrella. However, holding of umbrella produces a diminution of tasks in swinging and in another way using that for controlling another subsystem (umbrella)
- Each part of intelligence induces the storage board to reassign tasks with its intelligent priority.
  - Awareness: For eg. Provides awareness to the system about the cloudy or hot season.
  - Perception: For eg. Provides thinking or knowledge from the awareness.
  - Reasoning: For eg. Offers reasons from the decision network and evaluate true or false to help the knowledge and perception.
  - Learning: that plays a vital role in providing the knowledge-based analysis of the environment change and commit to maintaining the system stability by reassigning the tasks with the help of a super scheduler.
- Task mill schedules the inclusion and exclusion of tasks at each context level change.
  - Periodically checks and communicates with vital systems of intelligence and the status flags to detect a change in the environment to position itself for the addition or reduction of tasks.

5. CONCLUSIONS

This paper proposes the design of new intelligent real-time system methodology for the upcoming future where processor power, bandwidth of communication, increased architecture of intelligence are going to be plenty. These systems will strive to avoid meltdown or termination of systems under catastrophe arrivals and they try to manage critical catastrophe on an intelligent perception based scenario analyses and contain several damages like leakage of lethal gases and particles, detecting earthquake and taking measurement to minimize losses, recover unstable equilibriums of systems to minimize secondary and tertiary consequences to the extreme minimums that may be possible.

We firmly believe that our designed model can be used as a framework and applied to all kinds of large distributed real-time system in which intelligence and learning can be incorporated. It is evident from the studies carried out that each system should be understood thoroughly to infuse the properties of intelligence into it.

The work is under progress for further simulations of framework in order to determine the complexity of increasing intelligence. The details such as evaluation criteria, metrics, response control, result statistics, along with system performance measurements, etc. will be available in due course. The work could be extended by applying the basic tenets of the model to specific systems and by bringing out effective IRTS in the due course.

6. ACKNOWLEDGMENTS

The authors would like to thank every one of those experts within the framework of NDA (Non Disclosure Agreement) from industries and establishments across the world that provided us with information, practice drawbacks, lack of intelligence of particular systems currently in use.

7. REFERENCES

[1] Yun Niu, Guanzhong Dai, Reservation-Based State Feedback Scheduler for Hybrid Real-Time Systems, High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference, 25- 27 Sept. 2008, 198- 204.

[2] Insop Song, Sehjeong Kim; Karray, F. A Real-Time Scheduler Design for a Class of Embedded Systems, Mechatronics, IEEE/ASME Transactions, Feb. 2008, 36- 45.

[3] Liang-Teh Lee; Chia-Ying Tseng; Shieh-Jie Hsu, An Adaptive Scheduler For Embedded Multi-Processor Real- Time Systems, TENCON 2007 - 2007 IEEE Region 10 Conference, 2007, 1- 6.

[4] Celio Estevan Moron† and Hussein Zedan, Adaptable Scheduler Using milestones For Hard Real-TimeSystems,

www.cs.york.ac.uk/ftpdir/reports/YCS-93-191.pdf.

[5] Coskun, Erman and Grabowski, Martha, "Assessment of Intelligence Complexity in Embedded Intelligent Real Time Systems" (2002). ECIS 2002 Proceedings. Paper 46. http://aisel.aisnet.org/ecis2002/46

[6] Lopez, P.G.; Sandoval Gomez, R.J.; Torres, F.V., Measuring the efficiency of Schedulers for Concurrent Real- time Tasks in Uniprocessor Systems, Circuits and Systems, 2009. MWSCAS '09. 52nd IEEE International Midwest Symposium, 2009, 1078-1080.

[7] Hugo Marcondes, Rafael Cancian, Marcelo Stemmer, Antônio Augusto Frohlich, On the design of exible real-time schedulersfor embedded systems, Proceeding CSE '09 Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 02, Pages 382-387, IEEE Computer Society Washington, DC, USA ©2009.

[8] Richardson, P.; Sieh, L.; Elkateeb, A.M., Fault-Tolerant Adaptive Scheduling For Embedded Real-Time Systems, Micro, IEEE, 2001, 41- 51.

[9] C. L. Liu, James W. Layland, Scheduling Algorithms For Multiprogramming In A Hard Real-time Environment, Journal of the ACM (JACM) Volume 20 Issue 1, Jan. 1973, 46 - 61, ACM New York, NY, USA.

[10] Xu, Y.; Dong, Z. Y.; Zhao, J. H.; Zhang, P.; Wong, K. P.A Reliable Intelligent System for Real-Time Dynamic Security Assessment of Power Systems, Power Systems, IEEE Transactions, Issue: 99, 2012, 1.

[11] S. Ogrenci Memik , E. Bozorgzadeh , S. Ogrenci , Memik E. Bozorgzadeh , R. Kastner , M. Sarrafzadeh, A Super-Scheduler for Embedded Reconfigurable Systems, 2001.

[12] T.R.Gopalakrishnan Nair; A. Christy Persya, "Critical Task Re-assignment Under Hybrid Scheduling Approach in Multiprocessor Real-time Systems," Parallel and Distributed Computing and Systems, 2011. PDCS 2011, USA. 23rd IASTED International Conference on, vol., no., pp.130-137, 14-16 Dec. 2011.

[13] M. Vaidehi, T. R. Gopalakrishnan Nair, "Multicore Applications in Real-time Systems", Journal of Research & Industry, Interline Publishing, vol. 1, Issue 1, pp 30-35, 2008.

[14] D. J. Musliner, J. A. Hendler, A. K. Agrawala, E. H. Durfee, J. K. Strosnider, and C. J. Paul,The Challenges of Real-Time AI, IEEE Computer, Vol 28 #1, January 1995. Also appears as University of Maryland Technical Report CS-TR-3290 (UMIACS-TR-94-69).

[15] Kalinowski, Krzysztof, Cezary Grabowik, and Damian Krenczyk. "Predictive-Reactive Strategy for Real-time Scheduling of Manufacturing Systems." Applied Mechanics and Materials 307 (2013): 470-473.

[16] Ouelhadj, Djamila, and Sanja Petrovic. "A survey of dynamic scheduling in manufacturing systems." Journal of scheduling 12.4 (2009): 417-431.

[17] Burns, Alan, and Robert Davis. "Mixed criticality systems-a review."Department of Computer Science, University of York, Tech. Rep (2013).

[18] R. Alur, A. Trivedi, and D. Wojtczak. Optimal scheduling for constantrate multi-mode systems. In Proc. of the 15th ACM International Conference on Hybrid Systems: Computation and Control, HSCC '12, pages 75–84. ACM, 2012.

[19] A. Burns and T.J. Quiggle. Effective use of abort in programming mode changes. Ada Letters, 1990.

[20] P. Ekberg, M. Stigge, N. Guan, and W. Yi. State- based mode switching with applications to mixed criticality systems. In Proc. WMC, RTSS,pages 61–66, 2013.

[21] Burns, Alan. "System Mode Changes-General and Criticality-Based." WMC. 2014.