8/2/2020

**GROUP -K**

**Sumanth Mohan - C0752365**

**Narinder Pal Singh - C0751163**

**Assignment-2**

**Real Time Process Control Distributed System (Air Traffic Control Systems)**

**Abstract:**

**In this research paper, we discuss the concept of real-time distributed control system which is typically implemented by a set of computational devices (sensors, actuators, controllers, etc.) that run one or several tasks, which communicate data across a field level communication network. We discuss in details several use cases of the real time distributed control system using the use case and example working of the Air Traffic control system. The Distributed system has to be Implemented keeping several factors in mind. These factors could be the Design, patterns for the architecture, capacity planning, plans for automation, real time tracking and reporting, design and scaling of the applications. These factors have been discussed in the paper in detail. It should also be additionally noted that the reference papers we have used and referred in this report are professionally cited. The use cases have been Introduced and explained in detail for the air traffic control system which is a perfect example of the distributed systems framework. The use cases that have been described in this research paper are done after thorough analysis and research about the topics of distributed systems, air traffic control system and the real time process control systems.**

***Keywords: Distributed systems, Real Time Control, Air Traffic Control, control systems, power system, simulator, hardware and software open architecture***

**Introduction:**

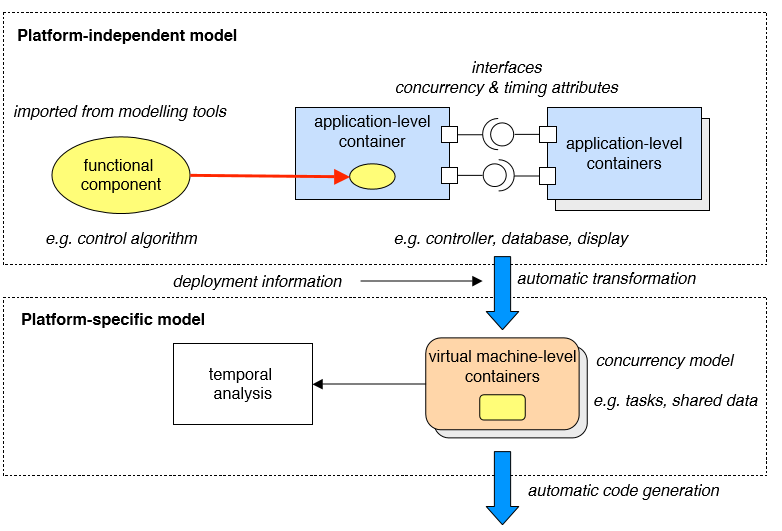
In this paper we are explaining the usage of large and complex real-time control systems being implemented on distributed hardware. The Real-time control systems play a critical part in the analysis, efficiency and the enhancement of the distributed system architecture ***(Reference 1, 2010)***[**1**]. Usually these systems process sensor input streams from the external environment and produce actuator outputs to perform closed-loop control functions. The control signals must be generated in a timely manner, that is to satisfy the requirements of real time. In addition, supervisory control includes control mode switching to manage various operating situations, detection of fault condition and recovery of failure, and the like. Due to the multi-dimensional interdependencies of the elements, these structures can become very difficult to maintain.

A real-time distributed control system is typically implemented by a set of computational devices (sensors, actuators, controllers, etc.) that run one or several tasks, which communicate data across a field level communication network (fieldbus). The successful design and implementation of real-time distributed control application requires an appropriate integration of several disciplines including control systems, real-time systems and communication systems ***(Research Paper 2, 2001)* [2]**

In order to achieve a successful implementation of real-time distributed control systems, it is imperatively important to derive and model all the timing constraints that the application must meet. For instance, the network can cause time-varying delays in the communication between field devices that can degrade the system performance, or the control computations competing for shared resources (scheduling problem) can cause unacceptable jitters for the control purposes, as it can be seen in . However, applying real-time systems methodologies, those time-variations can be assessed (determined or, at least, bounded). For example, with respect to fieldbus communication, a formal analysis and suitable methodologies have been presented in with the aim of guaranteeing before run-time that real-time distributed control systems can be successfully implemented with standard fieldbus communication networks.

1. **Design**

In this section of the report we discuss a very critical part of the real time process control systems which is the Design. In any Distributed system atmosphere, the design of the underlying components and the architecture plays a very critical part. The Real Time process control systems is an Integral part of the Distributed systems framework and works on enhancement of the distributed framework.

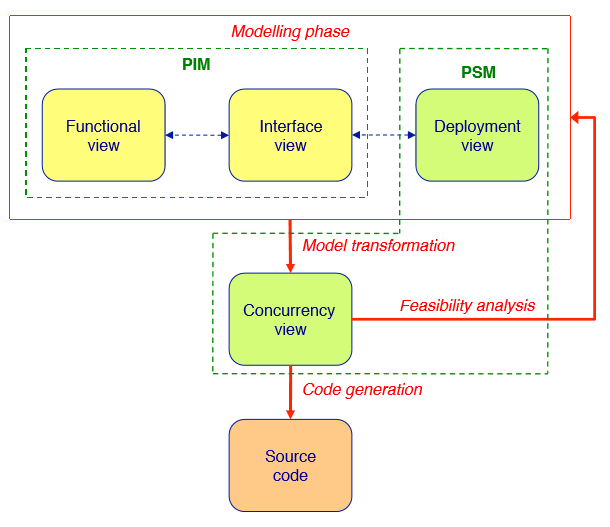


**Figure 1: Design of the Real Time process Control Systems**

System engineers, who are the process's end users, have complete control over the usable vision. We may use different notations to define the system's functional behaviour, as needed. Components of the interface view are classified as containers in which the functional code is embedded to provide device building blocks. The components of the interface view are called application-level containers (APLC) and are defined by the interfaces they provide and require that allow a complete system to be built as an aggregate of such components ***(Reference Paper 3, 2010)*[3]**. The components of the competitiveness view are executable entities that are generated automatically from phase views of modelling. They are called virtual machine-level containers (VMLC), since they depend directly on the VM 's mechanisms of competition, delivery, and real-time. Functional elements are embedded in VMLC, which provide the elements required to implement the speciﬁed temporal and distribution behaviour.

1. **Application architectures**

In this section we discuss the Application architecture of the Real Time Process control system. The architecture forms a very critical and very important part of the distributed system. It is the architecture which determines the overall outcome of the distributed framework.

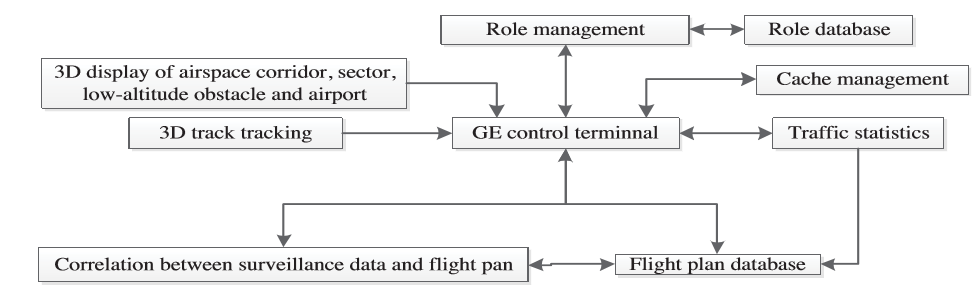


**Figure 2: Application Architecture of Real Time Process Control Systems**

Figure 2 depicts the application architecture of the Real time Process control systems. Communication drivers add basic communication capabilities to the real-time kernel and form the basis for the implementation of distributed systems. Indeed, the drivers have to follow the Ravenscar profile restrictions and be compatible with the real-time kernel to maintain the temporal properties of the applications using them. The AVM kernel has been expanded to support generic asynchronous serial lines and some other network interfaces. We may use different notations to define the system's functional behaviour, as needed **[3]**. Components of the interface view are classified as containers in which the functional code is embedded to provide device building blocks. The components of the interface view are called application-level containers (APLC) and are defined by the interfaces they provide and require that allow a complete system to be built as an aggregate of such components.

1. **Design Patterns for Scaling**

In this section of the report we discuss the Design patterns for scaling the real time process control systems within the distributed systems framework. We explain the design patters for scaling using an example of the Air Traffic control systems.

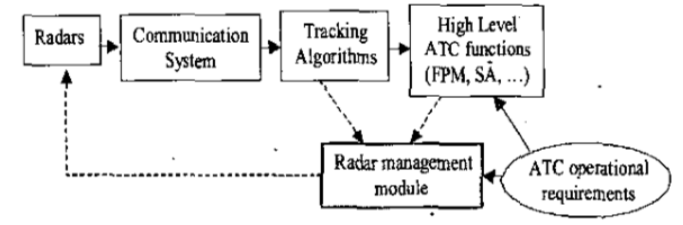


**Figure 3 : Design patterns for scaling of Air Traffic Control Systems**

Figure 3 Represents the Design patterns for scaling of the Real Time Air Traffic Control system which is a part of the distributed systems framework. The prototype system has eight key functions including task management, Google Earth (GE) control terminal, cache management, 3D monitoring, 3D flight plan, correlation between flight plan and surveillance data, traffic statistics, and 3D view of airspace and airport information. ATM controllers can access the network according to specific positions and authorities, and track and display path, sector and low-altitude obstacle information according to management criteria for airspace and airport layout. The 3D track monitoring function uses real-time message information from ADS-B, parses the flight call number, the address code and the time. The scaling of the components takes places as per the requirement of the distributed systems.

1. **design of operations**

In this section we discuss the design of operations of the Air Traffic control systems which is an example of the distributed systems logic.



**Figure 4 : Design of operations of the Air Traffic control real time system.**

Under this system, the radar or radar network providing aircraft trajectory observability is adaptively managed by the radar management module, based on continuous monitoring of the effects of high-level ATC functions, taking into account the operational requirements of ATC (also theoretically increasing in time) and the use of resources ***(Reference Paper4, 2019)*[4]**. The radars are controllable within this ideal framework: we can adjust the rate of track updates, the signal to noise ratio, ... To get predefined accuracy of the tracking device. In this contribution, we will concentrate primarily on Flight Plan Monitoring (FPM in Figure I), among all ATC functions of high level. Similar ideas to those outlined in this paper may be used in addition to controlling the sensor network allowing a robust aircraft separation assurance system (SA in Figure 1) or other functions. But, because of lack of space they won't be discussed in this article. It should be noted that there is a prerequisite under this architecture for perfecting adequate ATC which has robust monitoring. Therefore, the resource management subsystem would have to consider track continuity constraints in addition to the high-level ATC functions.

In order to achieve a successful implementation of real-time distributed control systems, it is imperatively important to derive and model all the timing constraints that the application must meet ***(Reference Paper 5, 2002)[***5]. For instance, the network can cause time-varying delays in the communication between field devices that can degrade the system performance, or the control computations competing for shared resources (scheduling problem) can cause unacceptable jitters for the control purposes, as it can be seen in

1. **Service Delivery**

In this section we discuss the Service delivery of the real time process control system. We take an example of the Air traffic control system to explain the service delivery of the real time process control system which in turn explains the distributed system logic. It is extremely challenging to construct even a feasible roster, with so many possible permutations of controllers and their required endorsements at the RTC ***(Reference Paper 6, 2017)[*6]**. Automation of the rostering process would have a positive effect on the ﬂight safety (because all operational constraints–like breaks–would be enforced) and help to keep track of controller qualiﬁcations and individual preferences.

**Input**

Given a set of airports with their opening hours and scheduled departure and arrival flights. We measure the overall traffic by the amount of movements that occur over a certain period (both arriving and departing flights).

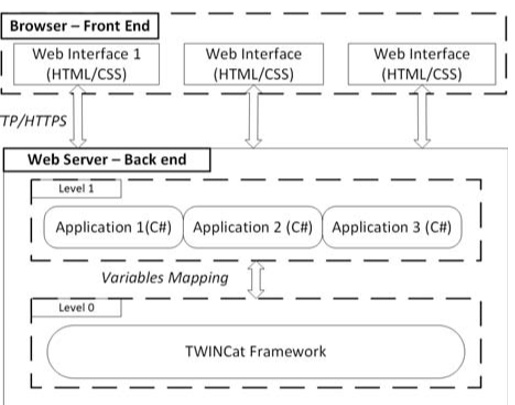
**Constraints**

Integrate the following safety and efficiency criteria for the service of RTC staff as constraints of our model: (a) Total number of airports allocated to one controller (b) Total number of movements per controller (c) Maximum number of controllers per airport (d) Possible scheduling conflicts should be avoided (e) Upper and lower bound to controller shift length (f) Maximum total time 'in our model. Distributing the total trafﬁc load between controller positions is the subject of sectorization research. The effective rostering of Air Trafﬁc Controllers is a complex and under-researched area of the personnel scheduling literature. ATC rostering inherits some features from the related staff scheduling problems, such as e.g. nurse scheduling

1. **Automation**

In this section we discuss the very Important topic of automation in the real time process control systems.

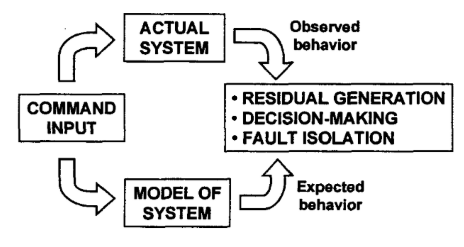
A specification for the software architecture for simulators is shown in Fig. 3 To get a better understanding of how software-level systems and technologies communicate. The software architecture used in this approach allows for the simulation in real time of discrete mathematical models of continuous and combinational systems, while submitting process-related data to a web interface ***(Reference Paper 7, 2019)* [7]**. A PLC is responsible for applying the Control Algorithm in this architecture. Various control methods can be used according to the simulated process. As can be noted, the architecture contains only the components of the process simulator and client applications. Since the PLC is an autonomous control body, the control logic can be implemented in any programming language, thus it is not included in the architecture itself. However, when the PLC is connected to the DAQ board, the PLC outputs in the Web server 's memory are associated with PLC variables. This interaction (mapping) is carried out using the TWINCat System, a specialised programme that Beckoff provides. The mapping is developed in an independent web server software application



**Figure 5: Automation in the Real Time Process Control systems**

1. **Monitoring Fundamentals**

In this section we discuss the very Important topic of monitoring the fundamentals in the real time process distributed systems. We once again take example of the air traffic control systems to explain this topic.



**Figure 6: Monitoring Fundamentals in the Air Traffic Control system**

With the command input and other system information, a model of the actual system being controlled is used to establish the desired behaviour. The difference between the model-based predicted behaviour and that of the actual system observed through a measuring device is quantified in a residual. The residual should be produced in such a manner that the greater the difference between the behaviours observed and predicted, the greater is the residual. Then, the decision-making process is needed to assess whether or not there is a fault based on the residual characteristics ***(Reference Paper 8, 2008)*[8]**. A fault is considered in the simplest form of decision-making if the residual crosses some predetermined threshold. Although more advanced techniques are widely used to mitigate false alarms (e.g. residual filtering, statistical and fuzzy decision taking, pattern recognition, etc.), all of them typically require some type of threshold check to assert a fault.

With the command input and other system information, a model of the actual system being controlled is used to establish the desired behaviour. The difference between the model-based predicted behaviour and that of the actual system observed through a measuring device is quantified in a residual. The residual should be produced in such a manner that the greater the difference between the behaviours observed and predicted, the greater is the residual

1. **Capacity Planning**

In this section we discuss the concept of capacity planning in the real time distributed systems.

There is no restriction of airspace per se, but the inability of sections of the ATC network to provide capacity at the rate of demand when such capacity is necessary ***(Reference Paper 9, 2002)[*9]**. This is due to the factors which follow:

* Fragmentation of the European airspace Staff
* shortages amongst some ANSPs
* Inappropriate staff scheduling and choice of airspace configuration at certain time periods - particularly on weekends

Clearly there is no “unique solution” to the problem of allocating capacity increases to individual ACCs so as to converge to an overall performance target. In order to ensure maximum overall cost efficiency. European system it is necessary to : In order to determine capacity targets for the Quantify the current capacity performance (baseline) of each ACC Build future traffic samples taking into account the current temporal demand profiles and the predicted growth provided by STATFO. The comprehensive set of simulation tools at the heart of FAP places it in a unique position to provide a greater insight into the effect on ATFM delay associated with future scenarios. The use of ATFM delay is proposed since it is a variable which is directly visible to the airline community and one upon which future strategy and investment decisions may be made. These performance predictions can address issues such as market access (difficulties of gaining airport slots) as well as delay predictions arising from evolutions in the level of capacity provision or the nature of the demand. Although such performance analysis for fbture scenarios is interesting in its own right, the quality of the predictions concerning traffic growth and capacity provision necessarily decrease as the prediction horizon increases. It is therefore more interesting and beneficial for the longer term to consider the sensitivity of the network to changes in the demand and capacity profile through comparative analysis

1. **Conclusion**

In this paper we have presented an integrated approach to the design and implementation of real-time distributed control systems. The Distributed system has to be Implemented keeping several factors in mind. These factors could be the Design, patterns for the architecture, capacity planning, plans for automation, real time tracking and reporting, design and scaling of the applications. These factors have been discussed in the paper in detail. The use cases have been Introduced and explained in detail for the air traffic control system which is a perfect example of the distributed systems framework. The use cases that have been described in this research paper are done after thorough analysis and research about the topics of distributed systems, air traffic control system and the real time process control systems

We have Enhanced our researched and explained the topic of Air traffic control in detail to give a better understanding of the working and the deployment of these control systems in the real time environment. The real time distributed control systems are a very critical component in monitoring the air traffic control systems. It should also be additionally noted that the reference papers we have used and referred in this report are professionally cited.

**References:**

**[1]**  Impact Analysis in Real-time Control Systems, Peter H. Feiler, Carnegie Mellon University, Pittsburgh, PA 15213 : <https://ieeexplore.ieee.org/document/792642>

**[2]** An Integrated Approach to Real-time Distributed Control Systems Over Fieldbuses, Pau Marti, Dept. of Automatic Control, Universitat Polithcnica de Catalunya -

**[3]** Design and Implementation of Real-Time Distributed Systems with the ASSERT Virtual Machine, Juan Zamorano , Universidad Polytechnical de Madrid - <https://ieeexplore.ieee.org/document/5641342>

**[4]** Design and Implementation of 3D Air Traffic Control Integrated Information Platform, Jinsong Wu, Southwest Sub-control Centre - <https://ieeexplore.ieee.org/document/8839544>

**[5]** Management of Radar Tracking systems for distributed systems, Juan A. Besad, Universidad Polit6cnica de - <https://ieeexplore.ieee.org/document/1174708>

**[6]** Scheduling Air Trafﬁc Controllers at the Remote Tower Center, Billy Josefsson, LFV Research & Innovation - <https://ieeexplore.ieee.org/abstract/document/8102018>

**[7]** Simulator for Automation and Control Systems in a Power System, Vasile Calofir, University Polytechnic of Bucharest - <https://ieeexplore.ieee.org/document/8744910>

**[8]** Conformance monitoring approaches in current and future air traffic control systems; Tom G. Reynolds - Massachusetts Institute of Technology - <https://ieeexplore.ieee.org/document/1052922>

**[9]** Capacity planning and performance predictions, Alan Marsden, Eurocontrol Experimental Centre, Bretigny sur Orge, France - <https://ieeexplore.ieee.org/document/1067931>