AdvancedComputingTechnologies Grid Computing and Expert System

Sanaa H. Abdulkarim Abushofa – Education College – University of Tripoli

Assma Sharaf m. Etakazi – Education College – University of Tripoli

Abstract

This search is broken down into two parts. The first part will examine air traffic control systems, the various scenarios which are present within them and the use of GridSim for the simulation of the system. The second part is regarding expert systems and their application within the real world.

Part 1 – Virtual Organization in a Grid Computing Environment

A) Analysis and Design Details:

A.1 Air Traffic Control System:

1.1 Introduction:

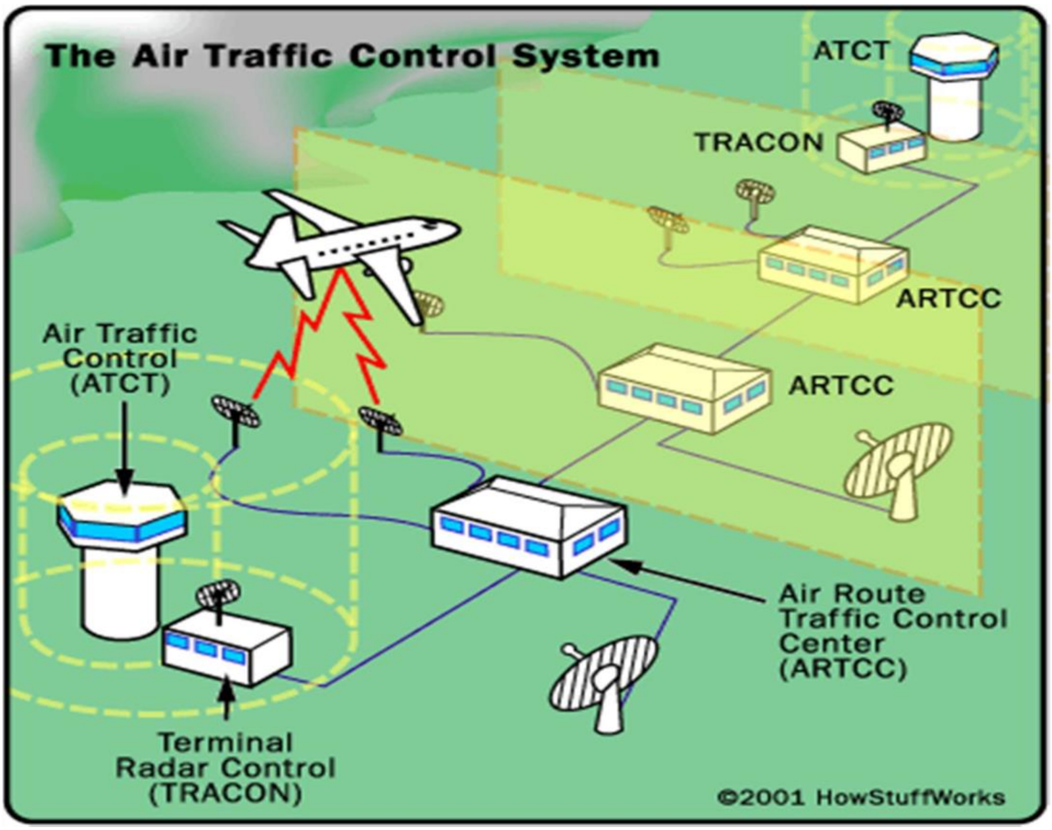


Figure 1 - The different aviation authority offices experienced by a plane amid its flight (Freudenrich, 2001)

As demonstrated in Figure 1, there are numerous positions in which the plane could be; every position is controlled by a various system. Freudenrich (2001) has divided the air traffic control system into:

- **Air traffic control tower** (ATCT): an ATCT is placed at each air terminal which has frequently planned flights. ATCT handle all departures, arrivals and ground traffic.
- **Terminal Radar Approach Control** called TRACON: its fundamental obligation is for ground flights, with the towers in the airplane terminal complex containing the framework telling the pilot when it is sheltered to begin moving the plane from the entryway to the NATCA (Controlled Movement Area).
- **Air Route Traffic Control** centres **(ARTCC) (En Route)**: there is one Air Route Traffic Control Centre for every inside. Every ARTCC oversees traffic inside all segments of its core with the exception of TRACON airspace and local airplane terminal airspace. This system is locked in after the plane has taken off and the plane is observed by a few Air Traffic Control Systems until it arrives at its destination.
- **Air Traffic Control System Command Centre** (ATCSCC): all ATCs are managed by this the ATCSCC, as well as overseeing the ATC inside the airport when any problems occur for example, bad weather, traffic jam, broken runway.
- **Flight service station (FSS):** the FSS gives private and smaller jets essential information such as weather reports, route information, terrain details and flight plan details, who are flying out of the smaller, more rural airports and countries. This is useful when smaller aircraft are in a state of emergency or co-ordinate search-and- rescue operations where the planes are not traceable.

1.2 Types of Air Traffic Controllers:

The air activity controllers are differentiated by obligation; the air controller's primary job is to keep the planes separate and every one following their own particular bearing. This guarantees the security of planes. The air activity controllers are set in diverse areas inside the air terminal while others are outside. Every kind of controller is in charge of a particular job and the following table will show every sort and their role (Conrad, 2010):

Table 1 – Controllers and their Role

NO	Controller	Role
1	**Flight Data Controller**	This controller is in charge of helping different controllers by giving and upgrading data on the weather and flight route details. This controller is in the control tower.
2	**Local Controller**	This controller is likewise in view of the control tower; they have the task of monitoring and verifying there is adequate separations between aircraft at the takeoff protocol.
3	**Ground Controller**	This controller is in charge of the wellbeing of all arrangements on the ground, utilizing ground radar to follow and track the arrangements alongside radio for reaching the pilot with directions.
4	**Departure Controller**	The primary obligation is to verify that there is a safe separation between the taking off of planes alongside data (bearing, and pace) for the pilot.
5	**Non-Radar Controller**	This controller helps the Radar Controller when there is a great deal of activity permitting it to guarantee the security and the stream of planes.
6	**Radar Controller**	This is concerned with guaranteeing that planes in flight are in full correspondence and that the pilot remains updated with data on headings, movement data, tallness, speed and climate.
7	**Approach Controller**	This controller is in charge of giving data to the pilot about the heading, rate and elevation of the aircraft.

1.1 Flying Stages:

Conrad (2010) and Freudenrich (2001) have shown seven activities which start with the pre-flight and end with the landing as indicated in Figure 2. The ATC facility is responsible for these activities.

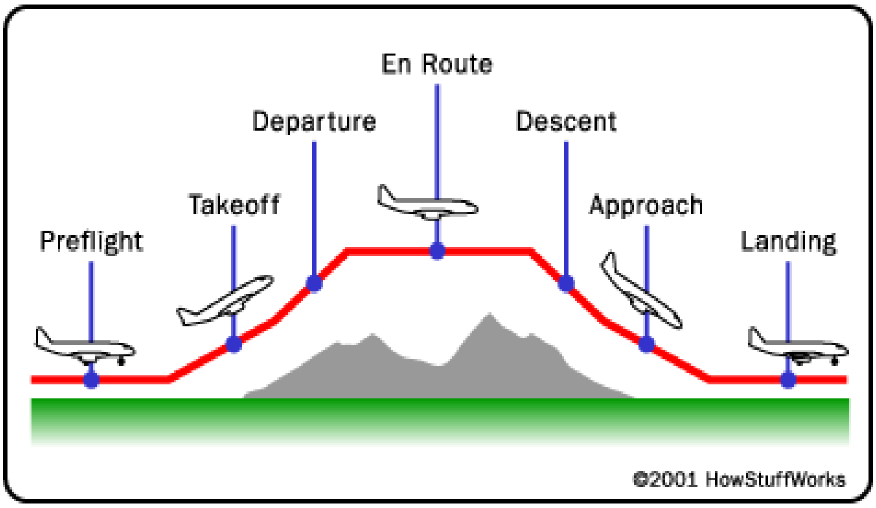


Figure 2 - Flying Stages (Freudenrich, 2001)

In this case, just six activities will be described in brief in the following figure 3 (Conrad, 2010):

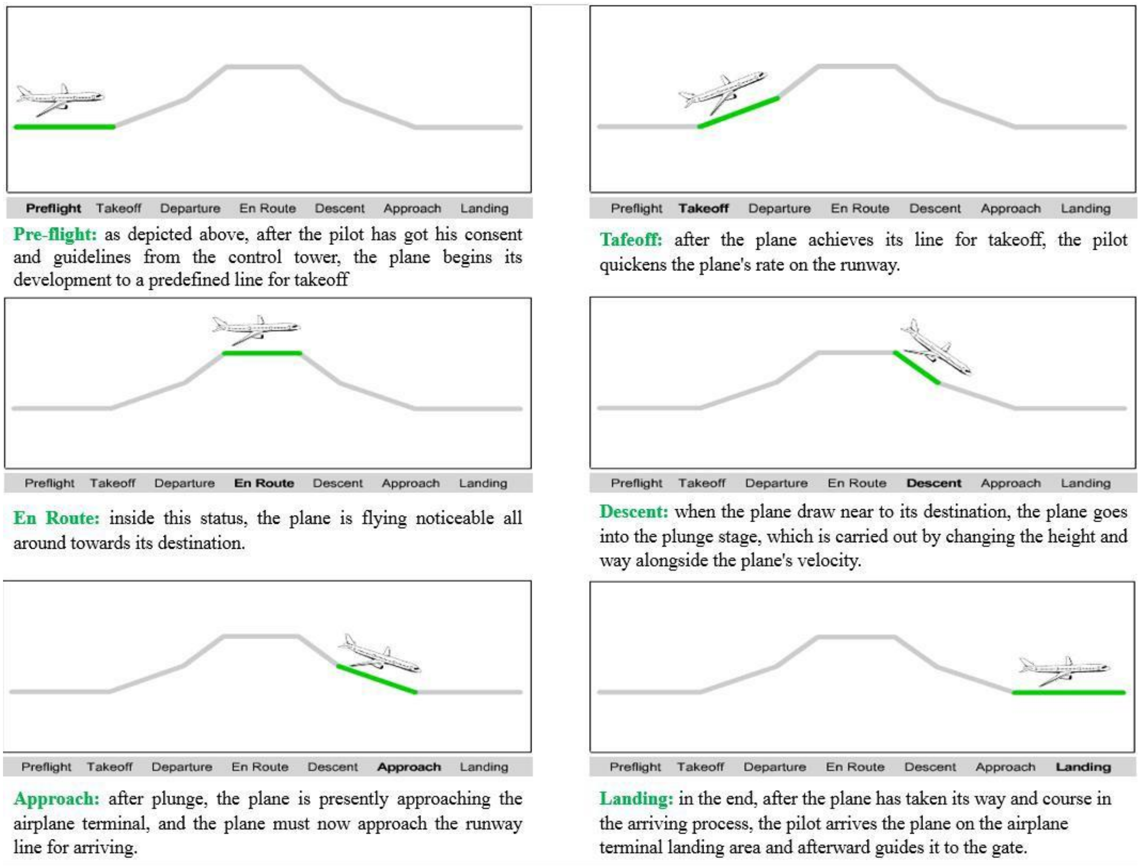


Figure 3 – Aircraft Activities (Conrad, 2010)

1.1 Air Traffic Control Scenarios:

4.1 Pre-flight and take off:

In this process, which is the first stage for any flight, the flight plan is checked by the pilot. When the pilot submits a flight plan, the Data Controller will affirm and sign the clearance. After that, in detail, the accompanying stages happen (Conrad, 2010):

1. The pilot must check and do a physical inspection of the plane, in order to ensure the pilot can be sure that the plane is perfect (free from operational and mechanical flaws).
2. The capacity of the flight's load, such as cargo, passengers and luggage are given to the pilots and the data controllers who must calculate these details with other conditions such as weather conditions en-route, the landing conditions at the destination airport and other such factors in order to calculate whether the flight is safe to go ahead or whether changes will need to be made.
3. At that moment, the pilot waits for the response from the Flight Data controller who accepts or rejects the plan. When the controller rejects the plan, the new plan will be sent to the pilot by the controller. However, when the plan is accepted, the pilot will receive the plan. Clearance is given alongside insights about the course the plane must take.
4. Then, the pilot will request subtle elements of the relegated entryway from

the door controller whose primary obligation is to relegate doors to planes. The allotted entryway number will go to the local controller to take into consideration freedom on the runway.

5. Then, the local controller will check for the measure of air activity and, taking into account the outcome. Data will be sent to the ground controller with the runway number. The ground controller will manage the pilot to the takeoff runway utilizing the radar.
6. The local controller in the tower now is in charge of utilizing radars to track and guarantee the security of the plane. Once the plane is allowed to take off, the local controller will give last freedom to taking off and set a radio recurrence for the plane.
7. After taking off, a TRACON radar, which is utilized by the flight controller, will screen the plane. In any crisis cases, the pilot will send a crisis sign to the TRACON which will send notice to the Flight Data controller to overhaul the plan. This means that a new flight plan is sent.

4.1 En Route and Descent:

The pilot now actuates a Transponder, which is a gadget used to identify approaching signs from the radar and telecast an enhanced radio flag toward that radar wave. The Transponder conveys the accompanying traits (flight number, height, velocity, and destination) to the Controller. On the Controller's radar screen a sound will be heard to the plane alongside its data so that the controller now can follow the plane on the radar screen (Freudenrich, 2001).

Once the plane takes off, the Departure Controller placed in the TRACON building keeps up a safe separation between planes at the airspace of the TRACON and the Departure Controller convey guidelines to the pilot with heading, speed and rate of rising. The Departure Controller screens the flight amid rising to the in transit bit, when the plane is out of the TRANCON airspace extend, the Departure Controller hands the plane to the ARTCC Controller as shown below (Freudenrich, 2001).

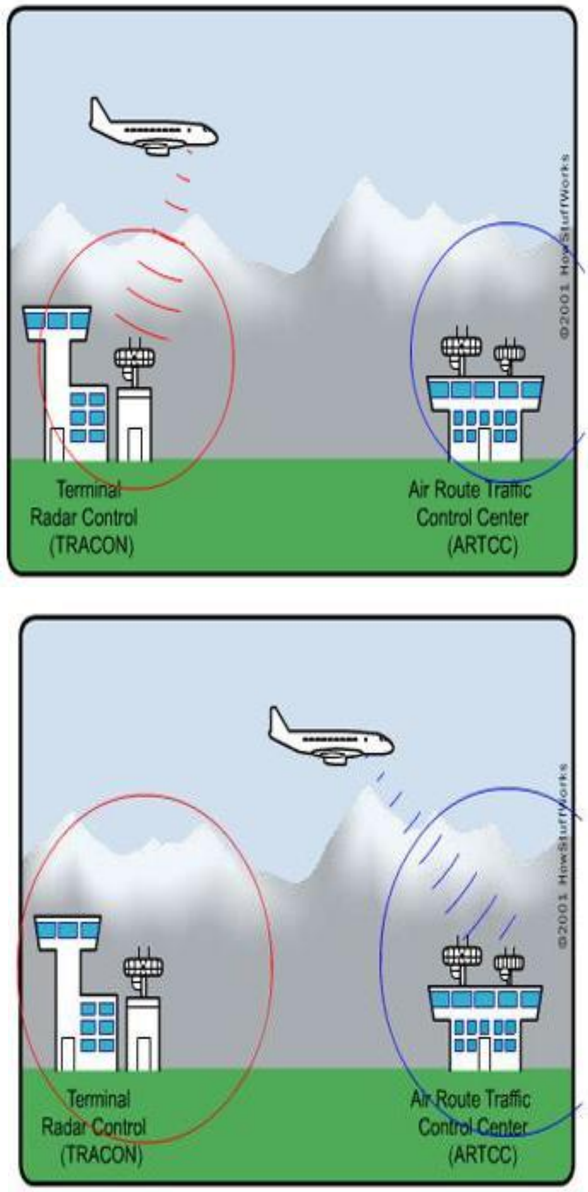


Figure 4 – How TRACON Works (Freudenrich, 2001)

Prior to the plane entering the ARTCC airspace, the Radar Associate Controller gets the flight arrangement's data, and once the radar is in the ARTCC, the plane is observed by no less than two air movement controllers. The Radar controller must manage all air-to-ground activities, keep up protected partition of airplanes inside the segment and directions exercises with different segments and/or focuses. The Controller just screens high height (over 24,000 ft.) and low elevation (underneath 24,000 ft.). The ARTCC Controller provides the plane pilot with the upgraded weather, air movement data, bearings of rate and height for every part. Until the plane leaves the area, the Controller passes the plane's data to the following segment before the plane enters the new part. The plane goes through a few divisions amid one flight, with the outcome that the controllers of every division may change the first flight arrangement in view of a few cases, for example, awful weather. This case proceeds until the plane arrives at its destination, when the ARTCC controller now coordinates the plane from a high elevation to a low height and begins its descent to the airplane terminal (Freudenrich, 2001).

At the point when the plane is approaching the airplane terminal, the controller offers guidelines to the pilot, for example, in connection to any alterations in course, speed and elevation, to place the plane in accordance with other aircraft. Contingent upon activity conditions, the controller may need to place the plane into a holding pattern, which is a standard course around the airplane terminal where the plane holds up until the air terminal can deal with its landing. The controller keeps on offering bearings to the plane pilot until the plane is inside TRACON airspace.

4.1 Approach and Landing:

At the point of arrival into the airspace of the airport, the aircraft must make use of the TRACON system. The aircraft must conform to all speed, pitch and line-up notifications which are issued by controllers in order to prepare for the landing using the standard methodologies. The pilot will adjust the aircraft for final approach and landing. When the plane has reached 10 miles from the air terminal, the local controllers must hand over aircraft to the air traffic control tower of the airport in order for them to manage their airspace effectively. Many tools such as surface radar are used in order to complete thorough checks of all surrounding areas to ensure it is safe to land planes on the runway and allow them to taxi to terminals. Once all checks have been conducted, the pilot is given permission to land. There are, however, occasions where pilots will have to enter into a holding pattern over the airport and wait for space on the runway to land. Furthermore, if the air traffic controller does not think it is safe to land after giving instructions to land, the pilot may have to abort the landing and try again (Freudenrich, 2001).

The local controller also has the task of ensuring that the weather conditions are acceptable for pilots to land while also ensuring that the distance between arriving planes is sufficient. Once the plane has landed on the runway, the air traffic controller must ensure that the pilot has directions to a taxiway and must ensure that the radio frequency is passed to the ground controller. Using ground radar, the ground controller can look over the runways and taxiways in order to ensure that planes do not cross runways to taxiways and do not interfere with other ground traffic. Once in this position, the local controller can now coordinate the plane into the air terminal for offloading baggage and persons (Freudenrich, 2001).

A.2 Data Flow Diagram for all ATC system activities:

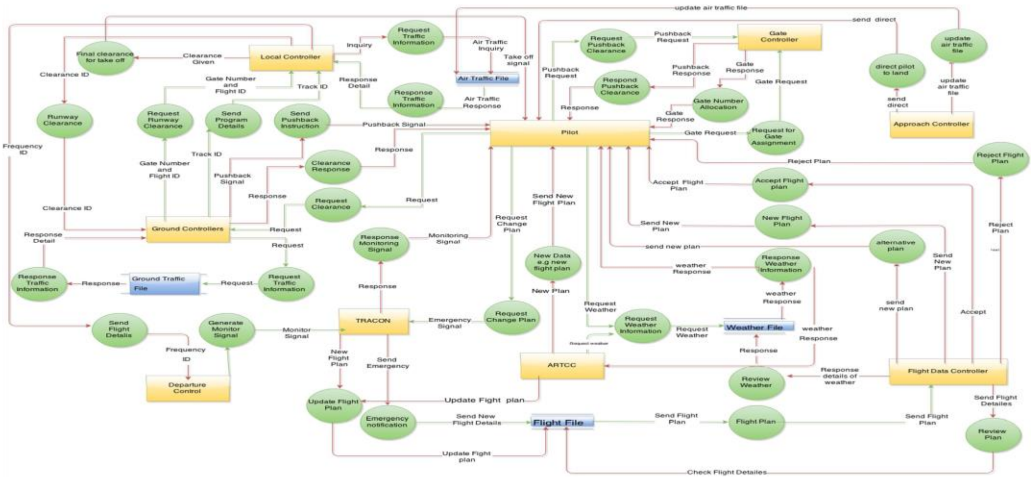


Figure 5 – DFD for Pre-flight, Takeoff, En Route, Descend, Approach and landing

A) An implementation of the ATC system in GridSim using JAVA programming language:

Introduction- GridSim:

The ATC activities can be implemented through GridSim, which is an open source java Application Programming Interface (API) under General Public License (GPL). Gridsim is utilized for scheduling resources and displaying them in distributed systems as well as supporting and sharing the available resources. It is similar to other Java software. In GridSim, varying entities and resources are supported such as single processor and multiprocessor environments for the purposes of simulation of the ATC system. Buyya and Murshed (2002) have described the system and entities that have been created by GridSim in the following table:

No	Entity	Description
1	Grid User entity	A client associated with the resources and GIS to submit or get Gridlets. The qualities of the Grid client include: client Id, name and Gridlet data.
2	Broker entity	The timetables grid resources package that holds the data for the jobs and gridlets.
3	Allocation Policy	The approaches that are utilized to designate resources
4	AllocPolicy	Handles the allotment approach of the inner GridResource.
5	GridLet	GridSim as name, length, size of file and size of output
6	GridStatistics entity	Holds measurable information utilized by different entities.
7	Grid Information Service (GIS) entity	It stays informed concerning accessible resources.
8	Input and output (I/O)	Date stream among entity represent (I/O).
9	PE (Processing Element)	Central Processing Unit (CPU) that serves to make grid resources.
10	Machine	Number of Processing Elements.
11	Millions Instructions Per Second (MIPS)	The unit of estimation of the Processing Element's speed.
12	Shutdown	An entity in charge of shutting all open entities methodically to demonstrate the end of the implementation.

Table 2– GridSim (Buyya and Murshed, 2002)

A.1 Creation of Grid users and Grid resources

In this case, Grid users and Grid resources are created with the implementation and initialization of GridSim. The ATC activities contain entities, such as the pilot, Ground controller and Data controller and files, such as flight plan file, weather and air traffic file. This means that Grid users represent entities and Grid resources represent files. The following five examples will describe how the ATC system implements the GridSim:

1.2 Example of GridSim:

Example one creates Grid resources:

In this example, the files / grid resources will be created for the ATC system's utilisation, such as the weather file, the flight plan file and air traffic file. With the revelation and initialization the GridSim processes Internal Packages and Classes will be demonstrated. One or more machines in the Grid resource might hold one or more PEs.

```
// Initialize the GridSim package
GridSim.init(num_user, calendar, trace_flag,
exclude_from_file, exclude_from_processing,
report_name);
```

The following is a brief summary of the steps expected to make a resource:

1. Make an object of the Machine List to save the machines which are identified with the Grid resource. The machine contains the accompanying variables: ID, the quantity of the PE and machine speed in MIPS every PE.

```
// 1. We need to create an object of MachineList to store one or more //Machines MachineList
mList = new MachineList();
//In this example, all PEs has the same MIPS (Millions
//Instruction Per Second) Rating for a Machine. int
mipsRating = 377;
mList.add( new Machine(0, 4, mipsRating)); // First Machine mList.add(
new Machine(1, 4, mipsRating)); // Second Machine mList.add( new
Machine(2, 2, mipsRating)); // Third Machine
```

2. Make an item to contain Grid resource variables, for example, system architecture, OS and list of allocation and machines.

```
String arch = "Sun Ultra";      // system architecture
String os = "Solaris";          // operating system
double time_zone = 9.0;      // time zone this resource located
double cost = 3.0;           // the cost of using this resource
```

3. A Grid resource object is made with the accompanying variables: resource name, correspondence velocity, and asset burden (amid top hours, occasions and off-peak hours and holiday day).

```
String name = "Resource_0";        // resource name double
baud_rate = 100.0;                 // communication speed long
seed = 11L*13*17*19*23+1;
double peakLoad = 0.0;        // the resource load during peak hour
double offPeakLoad = 0.0;     // the resource load during off-peak hr
double holidayLoad = 0.0;     // the resource load during holiday
                              // incorporates weekends so the grid resource is on 7 days a week
LinkedList<Integer> Weekends = new LinkedList<Integer>();
Weekends.add(new Integer(Calendar.SATURDAY));
 Weekends.add(new Integer(Calendar.SUNDAY));
```

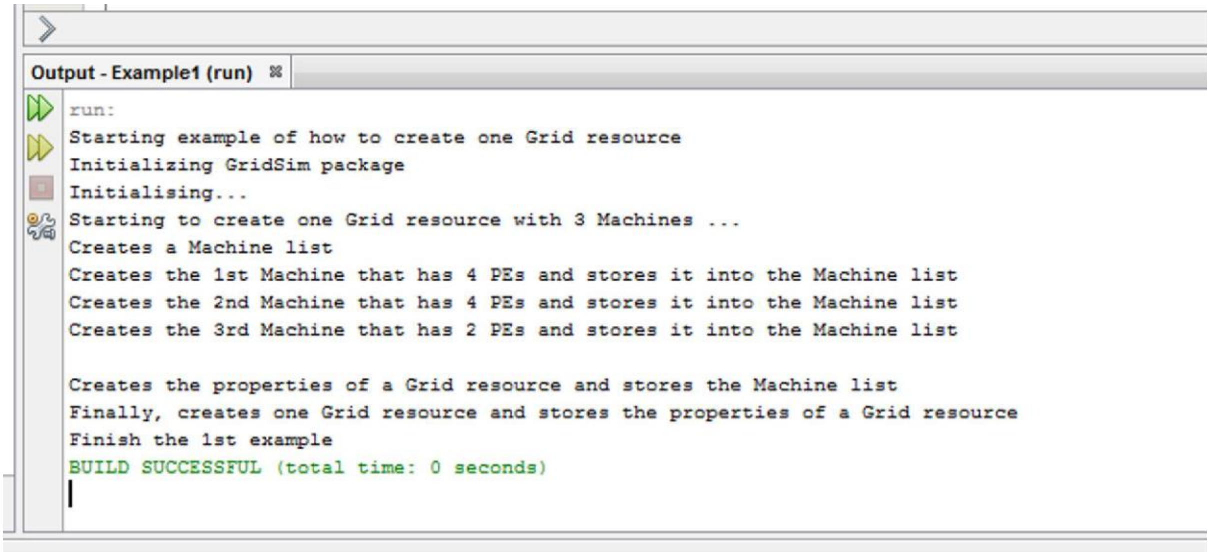


Figure 6 - Output

Example two creates Gridlets and Grid users:

In this example, the users within the ATC environment such as the Pilot, Ground Controller, Local Controller and others will be created. This means that the *CreateGridlet()* method will be utilised for creating the Gridlets. We will also make one or different Grid Clients (*CreateGridUser(GridletList_list*) and Gridlets. Every grid client has one or various Gridlets. In the example, arrangements of eight Gridlets were made. Five of these Gridlets were made utilizing the GridSim Random class while the other three were made physically. All of the Gridlets have the accompanying variables: user ID, length, size of file and output size:

```
// We create three Gridlets or jobs/tasks manually without the help
    // of GridSimRandom int
    id = 0;
    double length = 3500.0; long
    file_size = 300; long
    output_size = 300;
    Gridlet gridlet1 = new Gridlet(id, length, file_size, output_size);
```

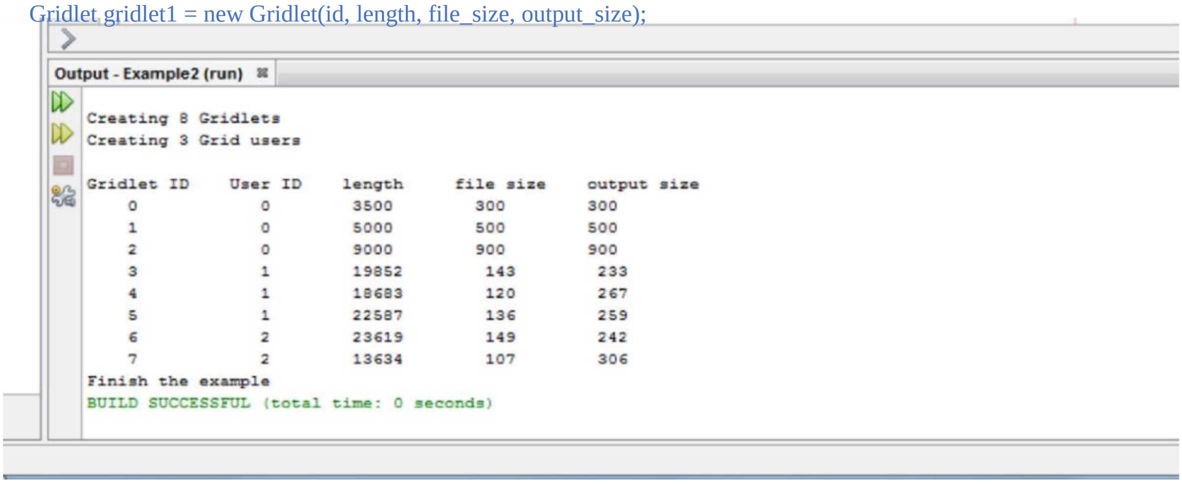


Figure 7 - Output

Example three creates and submits the Gridlets:

This example helps the ATC system to test comminution between users, such as from pilot to controller. Therefore, it demonstrates to make a test element and eight Gridlets; the eight Gridlets will be transferred to the test Entity, then the Test substance changes the Gridlet status to ―Success. The stages of the above stream can be depicted below:

1. A GridSim bundle is prepared and afterward introduced that makes a GridletList to store the Gridlets.
2. At that point it makes three Gridlets physically with their qualities (user ID, length, size of file, and size of output) the same in Example2.
3. At that point it makes five Gridlets utilizing GridSim Random and GriSim Standard PE.
4. A test GridSim entity is made.
5. After that, the SCHEDULE_NOW steady from the GridSim Tags class is utilized to transfer one Gridlet at once to the recently made Test element. The motivation is behind the GridSim Tags. SCHEDULE_NOW is to help control the postponement when sending Gridlets between elements.
6. The Test substance adjusts the status of the Gridlet to gridlet = super.gridletRecieve(); recieveList_.add(gridlet);

―Succes

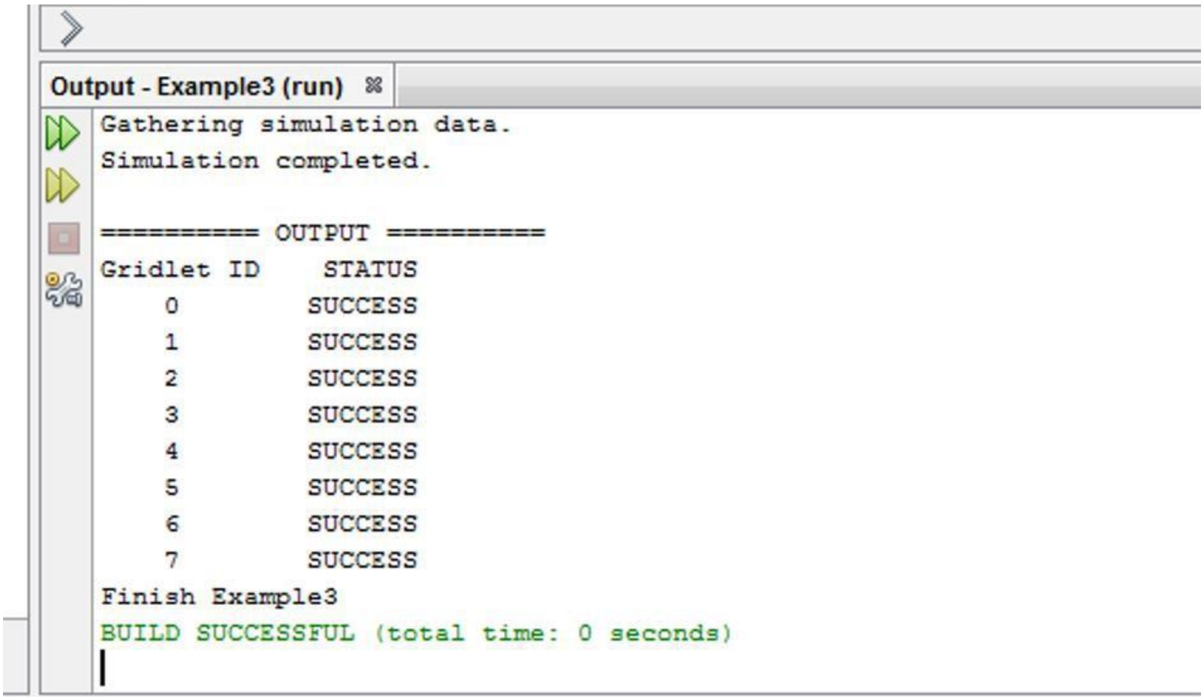


Figure 8 - Output

Example four submits the GridLets to the GridResource:

This example will represent how the user within the ATC environment submits a pilot to the resource, such as a flight plan or weather information. All controllers can check their respective files such as local controller and ground controllers. When it portrays how the grid client submits gridlets to a grid resource:

1. Eight gridlets were made and sent to the grid resource entity each one in turn.
2. Everyone is then allocated one-off identification details and afterward an appeal for the grid resource variables are made.
3. After it has been received, it is recorded. A circle is then made until each of the 8 gridlets are gotten and recorded in GridSim_stat.txt. The creation procedure of network asset and gridlets are the same as in Samples 1 and 2.

```
// Requests to resource entity to send its characteristics
super.send(resourceID, GridSimTags.SCHEDULE_NOW, GridSimTags.RESOURCE_CHARACTERISTICS, this.ID_);
// waiting to get a resource characteristics
resChar = (ResourceCharacteristics) super.receiveEventObject();
resourceName = resChar.getResourceName();
System.out.println("Received ResourceCharacteristics from " +
resourceName + ", with id = " + resourceID);
// record this event into "stat.txt" file
super.recordStatistics("\"Received ResourceCharacteristics " + "from " + resourceName + "\"", "");
```

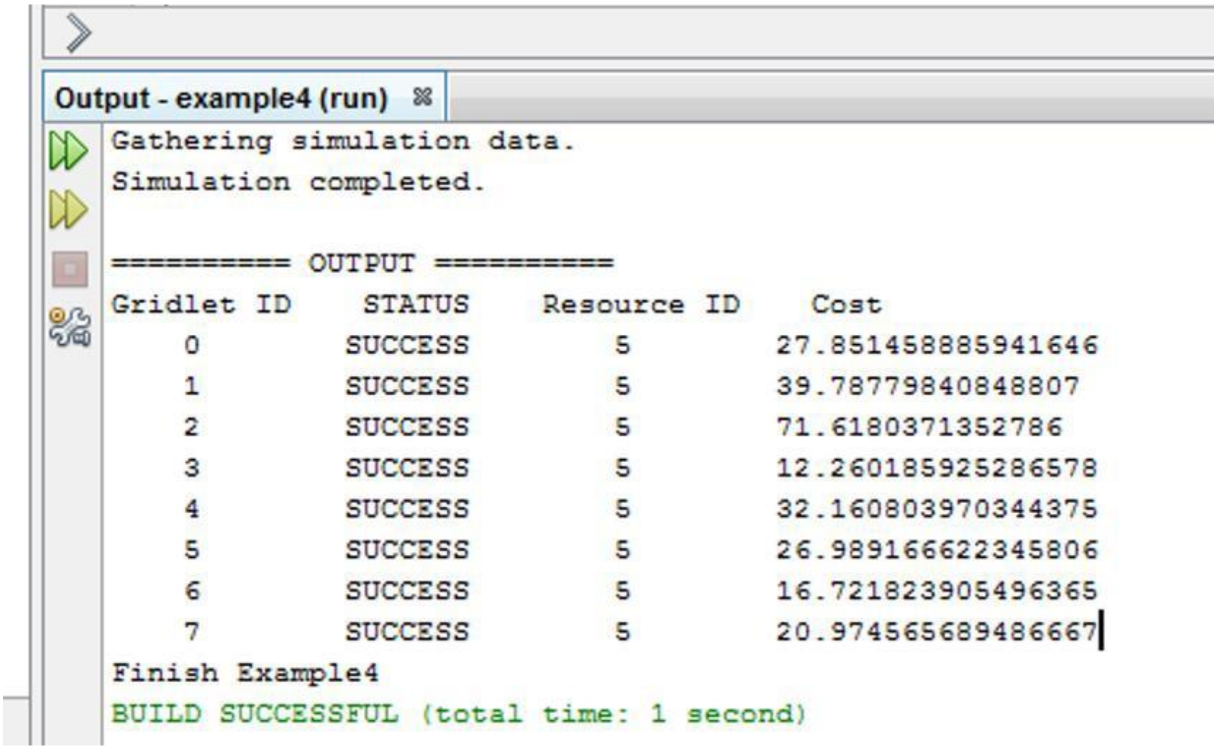


Figure 9 - Output

Example five submits GridLets to various Grid Resource:

This example will represent how the user within the ATC environment submits the same in the above example but in different resources, such as pilot can submit to one or many resources, such as a flight plan or weather information. When it indicates how the Grid client relegates his Gridlets to a Grid resources utilizing GridSim it:

1. Instates the GridSim bundle for simulation.

2. Makes one GridUser.

```
GridSim.init(num_user, calendar, trace_flag, exclude_from_file, exclude_from_processing, report_name);
```

3. Makes 3 GridResources:

```
createGridResource().
GridResource resource0 = createGridResource("Resource_0");
GridResource resource1 = createGridResource("Resource_1");
GridResource resource2 = createGridResource("Resource_2"); int
total_resource = 3
```

4. For every Grid Resource, three Machines are made with the accompanying variables: ID, number of PE and its speed in MIPS per PE.

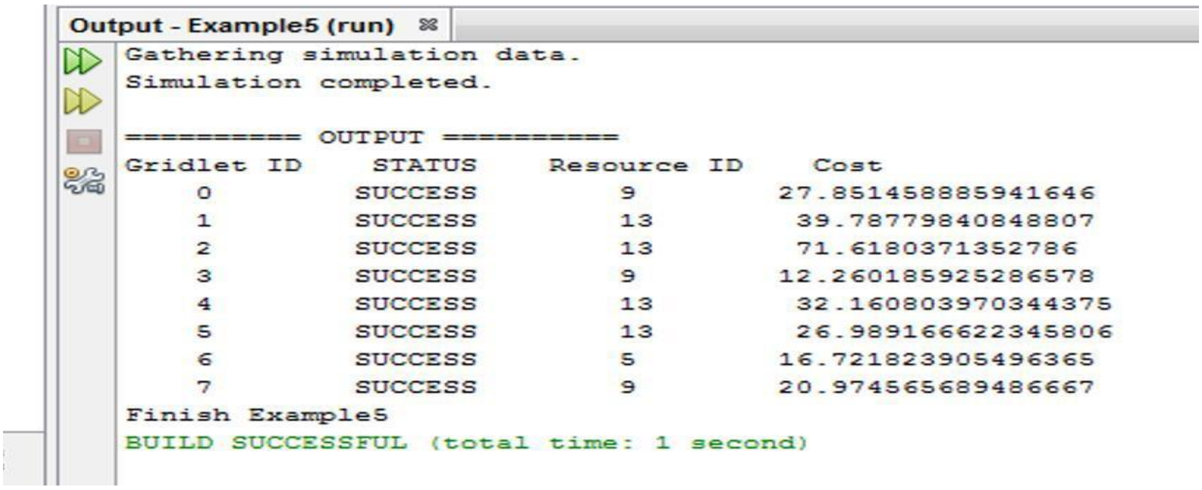
5. The Resource Attributes are allocated: system architecture, OS, time zone for resource spotted, and resource cost, such as example 1.

6. A short time later, eight Gridlets for the GridUser are made, three of them physically with properties: ID, length, size of file and size of output, and five of them with the assistance of the GridSim Random and GriSim Standard PE class.

7. The Broker appeals for a rundown of accessible assets with the accompanying variables: ID and Cost, which are enlisted in Grid Information Service (GIS).

8. After that, one must be patient for accessible assets and afterward it sends the Gridlets in arrangement to one of the three assets and it then holds up to get the Gridlet back after execution. Then, it save in Gridlet rundown object.

9. The asset then executes the Gridlet and changes its status to —Gridlet. Success


```
Output - Example5 (run)
Gathering simulation data.
Simulation completed.

========== OUTPUT ==========
Gridlet ID    STATUS    Resource ID    Cost
    0         SUCCESS        9         27.851458885941646
    1         SUCCESS        13         39.78779840848807
    2         SUCCESS        13         71.6180371352786
    3         SUCCESS        9         12.260185925286578
    4         SUCCESS        13         32.160803970344375
    5         SUCCESS        13         26.989166622345806
    6         SUCCESS        5         16.721823905496365
    7         SUCCESS        9         20.974565689486667
Finish Example5
BUILD SUCCESSFUL (total time: 1 second)
```

Figure 10 - Output

1.2 The ATC Activities in GridSim:

The following description will show how to implement the ATC system in the above five examples with more details when the ATC system starts with pre-flight and ends with landing with the suitable scheduling plicy.

For the emulation of the ATC Pre-flight and Take-off stages in utilizing Gridsim, it is first important to instate the Gridsim to set up nature for emulation. At that point a few Grid assets need to be made (Weather File, Air Traffic File, Flight Plan document and Ground Traffic record). After that, various Machine objects for every asset ought to be made and these ought to be held in the Machine List.

Each Machine contains various Processing Elements (PEs) or CPUs for disseminating the calculation force of procedures simultaneously with 377 MIPS every PE.

The first PEs or CPUs in the PC are good with the Grid resources which will execute preflight action utilizing the GridUser: Pilot and Gridlet: submit flight plan to Resource: Flight plan file until take-off, then the Flight Data Controller checks for Weather and Flight plan while an alternate controller checks Ground Traffic and Air Traffic records for every check. As will describe in the scheduling section, the time-shared scheduling policy is suitable in this ATC system consider the sensitive reasons

The second PE will make 3 different PEs to guarantee that it meets expectations all the while before the Pilot gets the Flight Plan endorsement. The correspondence between the 3 PEs and the first PE is through the info and yield information. The 3 PEs run Java Thread. At the point when one of the 3 PE completes execution, it will answer back to the first PE. The main PE will even now execute the capacities until the observing signs are produced in a typical circumstance to screen the airplane or crisis arrangement created in a crisis circumstance. The scheduling policy is time-shares policy.

The third PEs or CPUs in the PC are perfect with the Grid resource which will execute ARTCC airspace with the GridUser: pilot (transponder) and Gridlet: submit flight data to Resource: Flight plan File, with the controller now making air-to-ground correspondence, The third PE will continue working since it will continue taking care of data from the ARTCC radar to an alternate segment until it achieves the airplane terminal's TRACON radar. The scheduling policy will be space-share policy because it is suitable for this case as will describe in the scheduling section.

The fourth PEs or CPUs in the PC are good with the Grid asset which will execute Approach movement with the GridUser: Approach Controller and Gridlet: submit flight Instructions to Resource: Weather File, Air Traffic File and Ground Traffic File, with the controller now upgrading the majority of the records and setting up the plane for arriving. For every check, the fourth PE will make other 3 PEs to guarantee that it lives up to expectations all the while before the Pilot gets the Flight arriving regard and the PE will continue running until it achieves 15 kilometers from the air terminal's airspace. Therefore, the scheduling policy is time-share policy.

The fifth PEs or CPUs in the PC are perfect with Grid resources that will execute the Landing action with the GridUser: Ground Controller and Gridlet: submit Landing data to Resource: Air Traffic File and Ground Traffic File, with the controller now redesigning the greater part of the documents and get ready for arriving. At that point the User: Flight Data Controller checks for Weather and flight plan while an alternate controller checks: Ground Traffic and Air Traffic records. For every check, the first PE will make other 3 PEs to guarantee that it meets expectations all the while before the Pilot gets the Landing support. The correspondence between the 3 PEs and the fifth PE is through the info and yield information. The 3 PEs run Java Thread. At the point when one of the 3 PE completes its execution, it will answer back to the first PE. The fifth PE will even now execute the capacities until the flight is ceased at its entryway.

B.2 Scheduling Policies:

These strategies are utilized by schedulers to assign a few Gridlets to Grid resources. The goal of scheduling is to diminish the execution time of parallel Gridlets by submitting undertakings to machines to improve the framework (Multithreads). In GridSim, a single or multiple machines can be utilised. The machine comprises of one or numerous Processing Element(s) with its speed measured in MIPS or SPEC. The Grid resource can be a solitary processor with imparted memory (SMP) or a multi-processor dispersed memory which resembles a group that looks like a cluster (Buyya and Murshed, 2002). The scheduling policies are used to reduce the executing time and must be appropriate for demand situation. Consequently, the ATC system will used two types of the scheduling policy for implementation it as described in the section creation of Grid resources and Grid users.

The SMP is an asset involving a solitary machine which can be overseen by a **time-shared system** that uses round robin scheduling for multitasking. The aim of time shared systems is that each processing element is assigned such that the gridlets are receiving an equal share of processing elements.

The appropriated memory assets can be overseen by a **space-shared system** that uses a first- come-first-served or first-in-first-out asset distribution strategy. Unlike time-shared, in space- shared, processing elements are queued in order with gridlets and new processing elements are not able to use gridlets until such time that the previous have finished. There are many considerations when looking at air traffic control systems and it is therefore important to understand that the selection of the correct scheduling policy will define the way that one moves forward. Although the first come first served algorithm used with the space-shared system is useful as it is easy to implement, it can create bottlenecks and does mean that many processes will take much time in order to complete (Chandran and Balakrishnan, 2007). It is therefore advisable in this circumstance to use the round robin algorithm; although it is much harder to implement, it does not create bottlenecks and time-shared systems are used in substitute.

B.3 Discussion about Policy used for ATC System

With regards to the correct scheduling policy, there are many considerations for the ATC system. Firstly, it is of paramount importance that the safety of all activities is ensured and risk is as little as possible. All personnel need to have certain procedures in place to ensure the quick action when there is a state of emergency. Therefore it is important to use the correct scheduling policy to assist in such an event (Chandran and Balakrishnan , 2007).

Although the most common approach is the first come first served (FCFS) algorithm used in the time shared option as it is the easiest to implement, there are some drawbacks of this. Firstly, the system only works well with regards to a low number of processes occurring at any one time. This means that if there are a number of processes which are required to be done at the same time, the FCFS algorithm will not deal with them until the previous processes have been completed. This therefore has a great effect in the long run as it does not ensure safety.

The alternative option Ayaz et al. (2011) have shown is the round robin algorithm which is used with space shared resources. This is more versatile when compared with time shared as it allows for the algorithm to handle multiple processes at the same time and deal with multiple requests. Furthermore, it is also possible to ensure that time critical matters are dealt with swiftly as opposed to being in a queue of waiting processes.

B.4 GUI for an ATC System:

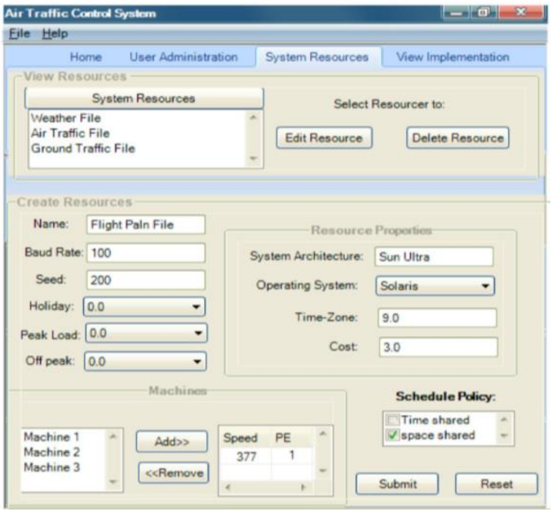
The ATC system requires entities which is users as well as file which is resources and the last is process that describe users will work on the resources. The following will show Graphical User Interface for the ATC System through GridSim:

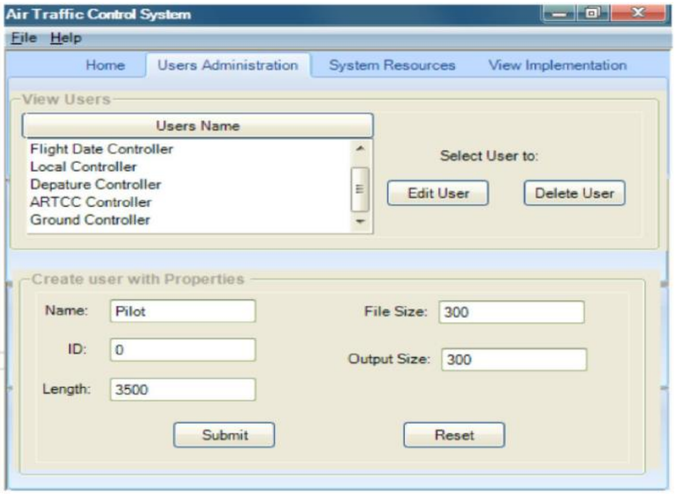
4.1 Home page:

The home page responsible to direct the users to different pages. In this page, user management bar will direct designers to create, edit and delete user page. The third is system resource bar which will direct to the resource management page. The last one is view implementation bar that will direct users to view all the users and resources has been created and all the submission:

4.1 User Administration:

In this page, Client is able to edit or delete created users and also can create new users and submit to the system.



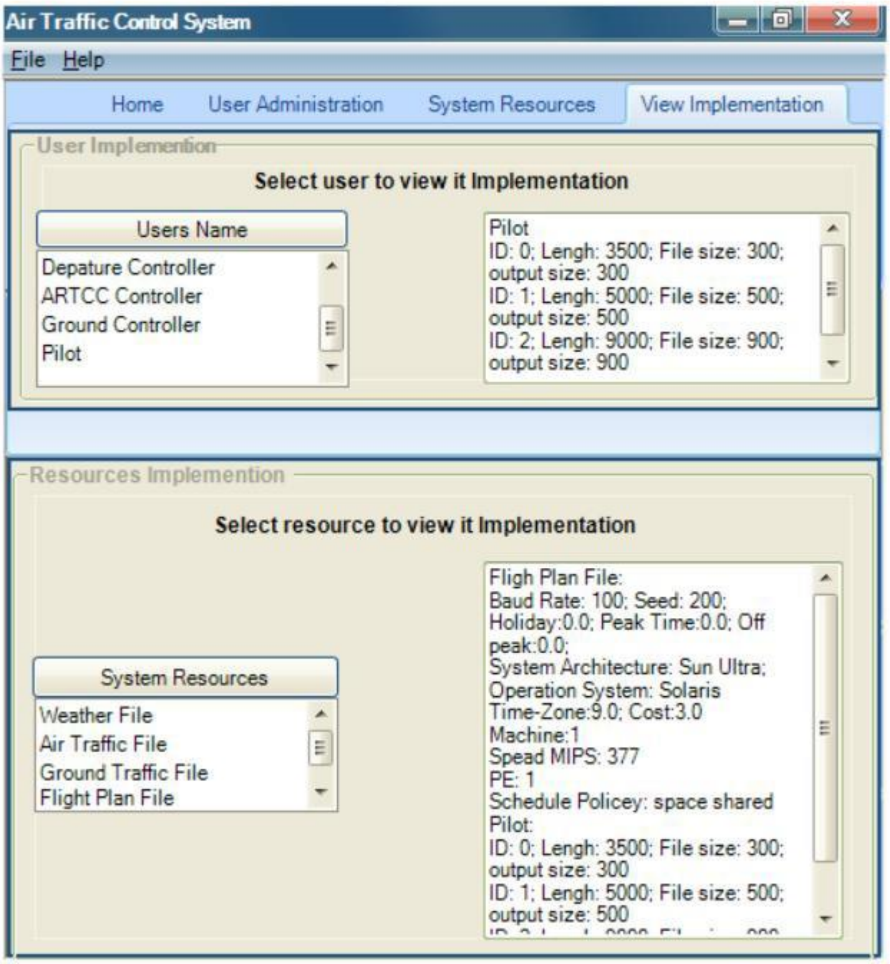


4.1 System Resources:

In this page, these fields are the attributes of the system to determine the performance of the ATC system. In these fields, the scheduling policy is an important which is time shared or space shared.

4.2 View Implementation:

In this page, clients are able to view all the users and resources that have been created. All the resources and allocated tasks will display in thi page.



Conclusion:

The Air Traffic Control system as an administration gives numerous preferences, which incorporate the safety and requests of flights as well as effectiveness, adequacy, association and administration of flights.

Using the simulation tool of GridSim, one is able to better simulate the activities of an airport by taking advantage of the various tools and offering then within the system. Using sets of entities and Gridlets to assist the program, the system can easily be replicated to near precision.

Throughout the course of this paper, we have discussed the various elements of the air traffic control system(s), investigated the flight scenarios and then applied these into the GridSim simulation system. Finally, a GUI has been implemented using the basic tools which were available.

This research has shown that the air traffic control systems are complex and in-depth systems which require a lot of attention not only from the controllers, but also from pilots, maintenance staff and system administrators. Should one link in the chain fail to work, the whole system is potentially in jeopardy

Part2 – Expert Systems andApplication:

2.1 Expert System

Expert Systems (ES) are transferring human learning into a machine, providing for them the capacity of taking discriminating decisions. ES is a limb of manufactured smart and was produced by AI group. This article concentrates on the writing survey of different articles about ES distributed between 1995 and 2004, the purpose for picking that dates on the grounds that the web was accessible to open clients after 1994. The paper uses reviews and characterizes ES systems utilizing eleven classifications which are

- Rule base framework
- Knowledge base framework
- Neural system
- Fuzzy expert system
- Object-situated strategy
- Case base thinking
- Modeling
- System structural engineering
- Intelligent operators
- Ontology
- Database approach

The papers has been composed by diverse clients originated from distinctive foundation, mastery this why you can see the creators shows up on a few written works distinctive philosophies and applications, while others have normal sort of system, for example, in guideline base framework, learning base framework and fuzzy logic. in this paper the vast majority of the articles accumulated were from distinctive classes, that gives a general thought of the fields that uses ES, and that doesn't imply that the fields that not said in this are not utilizing ES frameworks on the grounds that as a part of that time just these writing surveys were accessible .

There are numerous limiting issues portrayed in this study. This article in view of word base inquiry that implies there are numerous articles without a legitimate list or actually specifying ES in their record and that will avoid them from the indexed lists. An alternate restriction is society constraint as there is numerous articles written in different dialects and they may be vital for that field.

The meaning of ES philosophy is not finish as the overview barred the sociology. ES is interdisciplinary before perusing about ES you need to incorporate the frameworks that are identified with ES to improve thought of the capacities of ES.

2.1 Discussion of them

Rule Based System

Rule based systems are a piece of ES that gathers the information from a specialist human and characterize them into a machine in a manifestation of tenets. Case in point, on the off chance that you need to program a toy auto to push ahead until hitting something then regressive and advance once more. To place that in a structure principle advance in the event that you hit an article go places with regressive then move once more.

Rule based systems have been utilized as a part of numerous applications overall and probably the most imperative applications are:

1 Teaching

1 Electronic force arranging 1 Automobile force arranging 1 Advisory framework

1 Systemimprovement

1 Sensor control

1 Load booking

1 Geosciences

1 Knowledge representation

1 Production arranging

The diversity between rule based framework and object oriented frameworks in that there is no given request for the code execution. The purpose behind that is on account of the guideline based framework is situated of principles that contain expert's information and answer to some fundamental inquiries in AI.

Every standard has two side, left and right. The left hand side contains data realities and protests keeping in mind the end goal to execute them they must be genuine truths. In guideline based framework there is a motivation that incorporates all the standards each time a tenet executes effectively they will be expelled from plan as they are no more in need this procedure will proceed until there is no principles to execute. Standard based framework are actualized as single string program when all is said in done. The point of interest of tenet based framework is that in the event that we have an impeccable planned framework, the extricating and the executing of the specialists learning will be genuinely simple. Adjusting the framework, adjust any guidelines expected to be modified. Numerous rule based frameworks join rule managers which permit the tenets to be altered and changed effortlessly by those who are not experts.

Ruled based frameworks have numerous points of interest, for example, straightforwardness and homogeneity and that as a consequence of utilizing of IF-THEN guidelines sentence structure. Likewise, every guideline is autonomous so no stresses over where the new manage will be included the Rule base. An alternate playing point is that, the tenet base is divided from the surmising motor, therefore, the derivation motor and the learning base can utilize or work with other application parts, and that makes the framework more adaptable (Sasikumar M et al,2007).

Then again, notwithstanding these preferences, ruled based frameworks have a few disadvantages, for example, the absence of philosophy. Besides, the standard autonomy can be considered as drawback as well. Since, it prompted absence of intelligent in the middle of standards and that could result in inaccurate results (Sasikumar M et al,2007). There are numerous applications that taking into account principle based frameworks, for example, Psychiatric treatment, Biochemical nanotechnology, DNA histogram understanding and so forth (Shu-Hsien Liao, 2005).

Knowledge Based System

Knowledge Based Systems (KBS) are the comprehension and launch human learning by PCs. A framework which is assemble around learning base, putting away some information taken from master people into a PC. There is four fundamental parts for KBS : Knowledge base which contains learning about the field of premium, typically portrayed framework determination and it is the representation strategy for the information. Induction motor the center of critical thinking by doing that it will bolster the operation of different parts. Information building apparatus is in charge of planning and support of the ES, arrangement and formalization of strategies. It can likewise be utilized for speaking to a strategy utilizing thinking procedure. Executing the learning base framework. Particular client interface is the cooperation screen between the PC and client it can be utilized for dialog or any characteristic use (Barr,1981). Numerous applications executed by utilizing knowledge based framework, for example,

1 Medical treatment

1 Planner robots

1 Chemical procedure controlling 1 Decision making and learning 1 Chip outline

1 Case acceptance

1 Waste administration

1 Financial examination

1 Wastewater treatment

1 Urban design

KBS are keen data handling framework which they speak to space of enthusiasm by typical representation and they take care of issues by utilizing image control.

Neural Networks

Neural system of human body is extremely mind boggling, contains basic unites which are called neurons that each of these neurons have their capacity. By duplicating all the neurons capacities and actualize it in framework we can fabricate a manufactured neural system. We can utilize neural systems as a part of controlling development of robot in light of visual data or even personality perusing. For instance the astute vacuum cleaner which can clean the entire house and move around house by utilizing the same developments of human. An alternate utilization of neural systems we can utilize it for perceiving articles or visuals. Case in point, perceiving outward appearance or tone articulation and reacting utilizing all these data accumulated. This originated from learning process that really happens in neural frameworks. They can be considered unorganised functions (Buchanan, 1982). Neural systems applications:

1 Decision making

1 Robotic frameworks

1 Engineering ceramics

The structural planning has two fundamental classes:

- Feed-forward systems:

This structural planning also known as multi-layer building design, the first layer called data layer and the last layer called output layer or yield layer where different layers called shrouded layers, which normally comprises of maybe a couple concealed layers. The achievement for multilayer was the improvement of adapting by samples system. Number of neurons in concealed layers is dictated by the issue thought. The signs moves one path just from data to yield layer (Hajek M, 2005).

- Feedback or recurrent systems:

In feedback systems, signs can move in both headings by utilizing circles as a part of the systems. One of the fundamental peculiarities of this sort is effective and dynamic, additionally it has a solitary shrouded layer system.

2.1 Expert System Applications in Mining Industry

Expert Systems in mining are broadly connected. The primary preference of the product is the capacity to collect information, spare it for quite a while compass, upgrade the data. Along these lines, an organization can capacity regardless of the fact that the qualified authorities are not accessible. The spared information can be connected to enhance the capability of workers (Dolin, 1992).

There are 2 sorts of expert systems: static (Fig. 1) and element (Fig. 2). Static master frameworks are utilized as a part of those informative supplements where it is conceivable not to consider the progressions of world around happening amid the choice of an issue. The main expert systems which have gotten viable utilization were static (Konsho 2008).

The expert system has specific preferences, contrasting with the human specialists:

- It hasn't any prejudgments;
- It doesn't bounce to an inevitable end product;

These frameworks can work deliberately, consider all points of interest and pick the best probability.

The database can be huge. The information is embedded once and stays in the framework's memory forever. A human has restricted learning. In the event that some data wasn't utilized for quite a while, it is overlooked and lost; these frameworks don't supplant a human master, they are a device is his hand.

Dolin, G., 1992. What is ES. – Komputer Press, pp: 2.

Konshu, A.O., 2008. Concept of expert systems. Omsk.

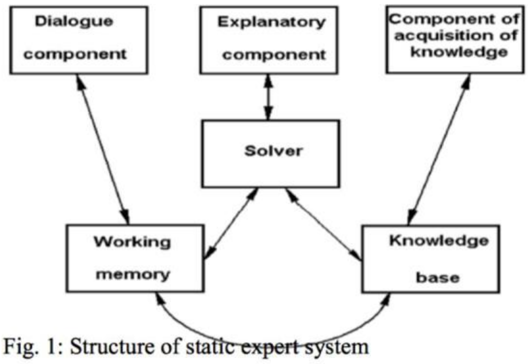


Fig. 1: Structure of static expert system

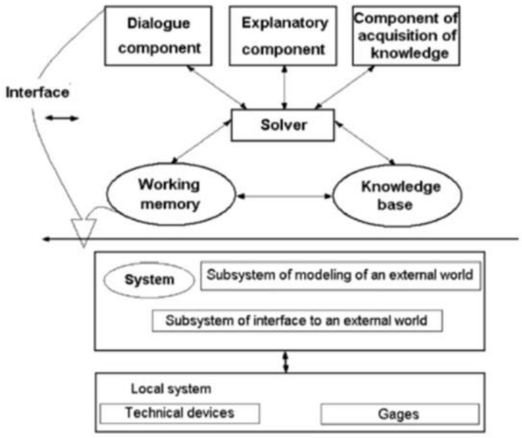


Fig. 2: Structure of dynamic expert system

References:

- Ayaz, S., Hoffmann, F., German, R. and Dressler, F. (2011). Analysis of Deficit Round Robin Scheduling for Future Aeronautical Data Link. *IEEE*.
- Barr, E. A. Feigenbaum, and P. R. Cohen. The Handbook of Artificial Intelligence.
- William Kaufman. Los Altos, CA. 1981 (3 vols.)
- B. G. Buchanan and R. O. Duda. Principles of Rule-Based Expert Systems.
- Buyya, R. and Murshed, M. (2002). GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing. *Concurrency Computat.: Pract. Exper.*, 14(13-15), pp.1175-1220.
- Chandran, B. and Balakrishnan, H. (2007). A Dynamic Programming Algorithm for Robust Runway Scheduling. *IEEEXplore digital library*.
- Conrad, L. (2010). *Air Traffic Management System*. [online] Virtualskies.arc.nasa.gov.
- Available at: http://virtualskies.arc.nasa.gov/atm/2.html [Accessed 1 Mar. 2015].
- Dolin, G., 1992. What is ES. – Komputer Press, pp: 2.
- Freudenrich, P. (2001). *How Air Traffic Control Works - HowStuffWorks*. [online]

HowStuffWorks Availableat: http://science.howstuffworks.com/transport/flight/modern/air-traffic-control.htm [Accessed 10 Mar. 2015].
- Konshu, A.O., 2008. Concept of expert systems. Omsk.
- J. Bachant and J. McDermott. "R1 Revisited: Four Years in the Trenches."
- Shafeeq, A. and Saxemn, V. (2008). DESIGN OF FORMAL AIR TRAFFIC CONTROL SYSTEM THROUGH UML. *Ubiquitous Computing and Communication*, 6(3)
- Nolan, M. (2011). *Fundamentals of air traffic control*. 5th ed. Belmont, Calif.: Wadsworth Pub. Co.
- Report STAN-CS-82-926. Stanford University, August, 1982