# Chapter 1.

# Flight simulators.

## Flight simulators.

### Basic description of flight simulators.

A **flight simulator** is a device that artificially re-creates aircraft flight and the environment in which it flies, for pilot training, design, or other purposes. It includes replicating the equations that govern how aircraft fly, how they react to applications of flight controls, the effects of other aircraft systems, and how the aircraft reacts to external factors such as air density, turbulence, wind shear, cloud, precipitation, etc. Flight simulation is used for a variety of reasons, including flight training (mainly of pilots), the design and development of the aircraft itself, and research into aircraft characteristics and control handling qualities.

Flight simulators are commonly used to maintain level of pilot training. Several different devices are utilized in modern flight training. Cockpit Procedures Trainer (CPT) are used to practice basic cockpit procedures, such as processing emergency checklists, and for cockpit familiarization. Certain aircraft systems may or may not be simulated. The aerodynamic model is usually extremely generic if present at all.

The one of main goals of training on flight simulator is to increase highly automated skills and quality execution but non-standard solutions, which are characterized by flight activity during non-standard situations during flights, the load on intellectual pilot function while performing assigned tasks kinds of aircraft.

### Types of flight simulators.

All available aircraft simulators can be divided into two main types:

* software simulators;
* training complexes.

In modern terminology, aviation simulators with a fixed cabin belongs to flight simulators. Simulators significantly differ in design depending on the destination: from mechanics and electronic equipment from the dashboard and the front part of the fuselage, designed to train pilots to computer programs PCs. Many software simulation realism is characterized as low because it does not allow the use of all the senses and is used in gaming purposes for personal computers.

Software simulators divided into procedural and comprehensive. **Procedural** aircraft simulator designed for training flight crews. This technical teaching tool that allows you to shape the skills needed in the real world. It has the following main features: simulator imitation of individual fragments of conditions of real activity pilot; the possibility of practicing in certain operations and actions of real pilot with cab equipment; the possibility of objective monitoring results of all operations, practiced on the simulator and instructor’s actions.

The procedural simulators provide training of specific actions, such as control of the aircraft, engines and aviation systems, staff, management of electronic equipment, combat use and so on. Typically, this kind of simulators are composed of display boards and instrument simulators and simulators control levers, whose boundary movements, load characteristics and tactile sensations correspond to real at all stages and modes of flight. Some devices that are closest to the operation are real.

The procedural simulators designed for working crew procedures and training for the flight. Purpose consoles, instruments and controls are generally simulated using touch monitors. For the convenience of individual panels and controls can be presented as full-size models. Additionally, depending on the amount of realized tasks, training can be divided into the following types:

1. Functional (primary) cabins, which are modeled to display information controls. They make it possible to deepen the knowledge of students-pilots of aerodynamics and aviation equipment, off procedure during the pilot operation of aircraft. Primary aviation simulators are usually the simplest, often made by the aviation units and schools. Stands and models can be considered as functional simulators.
2. Specialized training designed to prepare cadets-pilots for doing specific activities, for development of certain psychological qualities and skills of action in special cases in flight.

A **comprehensive** aviation training simulator implements similar to procedural simulator, but on advanced level and has such basic features as approaching the maximum conditions of the pilot in the simulator to the real conditions of the flight. Providing practice on the simulator in general of all tasks of a real pilot, which he carries in flight; enable objective monitoring results of all tested tasks.

An integrated simulator - the highest level of technical training to prepare flight crews and effective means of maintaining trained skills of pilots. An integrated simulator recreates real cab interior also makes it possible to work out all modes of operation of the aircraft. Simulators of highest qualification level have complete set of tools that provide adequate performance in all channels of perception cadet.

### The Concept of Real-time Simulation

Typically, office operating systems provide acceptable control of the mouse and a mechanism for ensuring that the software is activating mouse 50 times a second part of the operating system, and is transparent to the user. The fact that the operating system does discretize this time, in a minor steps, so that the mouse control code is guaranteed to be performed 50 times per second. These time steps less than we can discern with our eyes (and hands and brain), giving the mouse that appears to be instantaneous and continuous movement, in fact implemented at discrete intervals.

Of course, we are all accustomed to such systems. The cameras at a football game capture frames every 1/25 second, which are transmitted to the TV in the house. Because the time step is so small, it seems that the players on the field are moving in the normal continuous mode, we would expect to see if we were sitting on the bench in the match.

The same situation occurs in the simulation flight. On the plane, the pilot moves a steering wheel. Assuming that a direct cable connection to the control surfaces (and not paying attention to the inertia of the control surface), the elevator moves at once, causing a disturbance to the plane, which is considered as a change in the pitch of the pilot, which reacts with the other control column movement for the correction of the pitch angle. In flight simulator, the position of the handle is digitized, elevator deflection is calculated, the new pitch ratio is computed and the image displayed on the screen via the visual system with a new pitch, allowing the pilot to adjust the attitude of the aircraft. The important point is that the total time for this calculation should be sufﬁciently short, so it seems the pilot instantly. In today's simulator, these calculations should be completed within 1/50-th of second or 20 milliseconds.

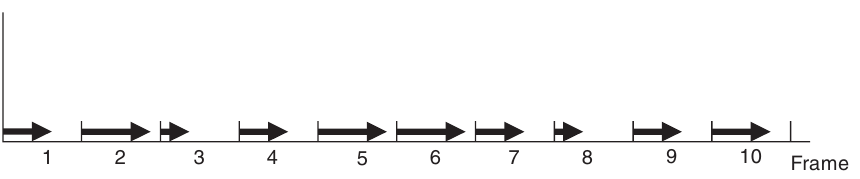
This concept is illustrated in Figure 1.1, which depicts 10 frames of a simulation. The arrow for each frame shows the proportions of the frame used in the calculation of the simulation. If the frame time is sufficiently small, say 1 / 50th of a second, and if any frame in the computation will never exceed the time frame, the real-time simulation.

Fig. 1.1. Real-time frames.

Note that there is an important distinction between the real-time calculations and fast computers. Although real-time simulations may need a fast computer, all calculations have to be completed within the time frame (Burns and Wellings, 2001), while for the fast computational problem, only metric is the overall time. Some simulation packages assume that the code generation in a compact form allows the real-time computation. While this may be true for certain applications, in all examples described in this book, the term real-time is used to indicate that all calculations are resolved within one frame for each simulation frame.

Another important factor has been affected. In addition to the basic modeling tasks, the processor may also be required to perform other background tasks. If this additional load calculation results in contradictory within a frame period to perform this task of modeling, simulation in real time cannot be sustained. Therefore, the operating system is an important part of any real-time simulator. The operating system must ensure that it will perform the task of simulating each frame and never introduce delays that cause the problem of modeling a frame exceed its limit.

The safety critical real-time environment, it is necessary to demonstrate that the frame rate in real time can never be broken beyond any reasonable level of doubt. Although the software for flight simulation is not critical for safety, real-time limit still to be performed mode. As a rule, it is the duty of the simulator of the designer. You can monitor the performance of real-time flight simulator and record any violations of the frame period. However, if the frame rate does fall, it is usually apparent to flight crew, as it is noticeable discontinuities in the visual system, or discernible lag in response to the aircraft or even changes in the frequency of the sound output system.

The simulator designer has, in fact, the time budget for the completion of all calculations in the frame and, therefore, trying to use as much of the frame time as possible, as it may leave a small margin for error in these estimates (or for future expansion), in particular as some calculation times data dependent. Taking into account all restrictions on the content of the visual system of the scene, processing flight model, engine model, the weather and so on. It is not uncommon for a frame period should be exceeded sometimes even for full flight simulator, in particular, as the simulator manufacturer cannot have full control over the behavior of the video card in all conditions of flight. However, providing real-time operation, especially for the worst conditions, it is an integral part of validation and acceptance testing system.

### Training versus Simulation

Flight simulators are used in flight training and is easy to assume that the two terms are synonymous and interchangeable. However, light training provided fulfil l education requirement. The equipment used for this preparation may also include a light simulator, but in this role, just equipment simulator. Simulator, together with an instructor and curriculum is a training package. Confusion has arisen because in some cases, the flight simulator is a pure replacement of the aircraft, and this may have led in the past to purchase flight simulators that were poorly adapted to the requirements of vocational training and is therefore provided a low education.

The first phase of purchases for any flight training program is an analysis of training needs. This establishes that the training program is required, and what is required. On one hand, if the training equipment low, it cannot provide effective training. On the other hand, if the inflated preparation equipment, cost simulation equipment may be excessive.

The often-quoted example of an effective coach is a "cardboard bomber". In the early stages of flight, pilots have to learn a series of checks, including checks before the flight, in-flight inspections, and emergency shutdown test. Sitting in a normal chair, in front of a cardboard facsimile of the cabin, with no moving parts whatsoever, the pilot can indicate on the tool, or press the switch that corresponds to each inspection. Photos used in such a way that each item physically resembles the actual equipment, and is located approximately in the same place as in an airplane. However, the requirement of training equipment just to help pilots remember checks. It is obvious that such equipment would be inappropriate in the later stages of learning, which need real tools aircraft or switches. Similarly, the actual use of aircraft equipment would be an unnecessary expense at the initial stage of training.

For the core curriculum, analysis of training will be conducted by a team of professionals with a good understanding of fl ying training, simulation technology, teaching methods and human factors. The team will not include flight instructors, which will give further training or simulator company that will produce the equipment. For the airline or military training, specific training will be ed in terms of the desired results (or output) training. For example, the tool for the coach, the requirement may declare "the completion of the training, the pilot must be able to demonstrate ILS approach with a cross wind to the maximum permissible value for the aircraft with one engine failed," as part of the training requirement. The purpose of this single statement is that the simulator requires ILS instrument, realistic model of an engine failure, the model forecast, which includes cross-winds and presumably a database of navigation aids including ILS beacons. Please note that the requirement does not include the elements of motor control, the details of weather patterns or the maximum number of ILS approaches must be provided.

Presumably, such information would be given elsewhere in the training requirement. The training needs analysis team will review the training syllabus, discuss the detailed requirement with the customer (possibly to modify or clarify the requirement), review the technical options (to establish any constraints or parameters) and discuss the training methods used by the instructors. For example, some of the questions to be asked about the ILS simulation might include the following:

* The need for aural cues;
* The accuracy of the simulated ILS;
* The range of the ILS;
* The failure modes of the ILS;
* The method of selecting an ILS frequency;
* The number of ILS channels;
* The failure modes that can be set by an instructor;
* The physical representation of the ILS.

This ﬁnal consideration might give the option of using a graphics display to represent the ILS instrument rather than a mechanical emulation, leading to the clariﬁcation of further issues, including

* The resolution of the ILS display;
* The need for anti-aliasing;
* The update rate of the display.

The important point to bear in mind is that the training requirement extends across the whole scope of the training programme, and even includes non-functional requirements, such as

* Access dimensions;
* Power requirements;
* Emergency lighting;
* Reliability and availability ﬁgures;
* Air-conditioning;
* Safety issues.

Once the training requirements are clearly deﬁned, these are passed to prospective manufacturers who will be invited to tender to supply the training equipment. There is often some variability in these requirements. For example, the requirement may simply state the tasks to be trained and the level of skill to be attained, and possibly the time available to attain it, using the training equipment. In this case, the simulator manufacturer can match their equipment to the training requirement, advocating one technology rather than another.

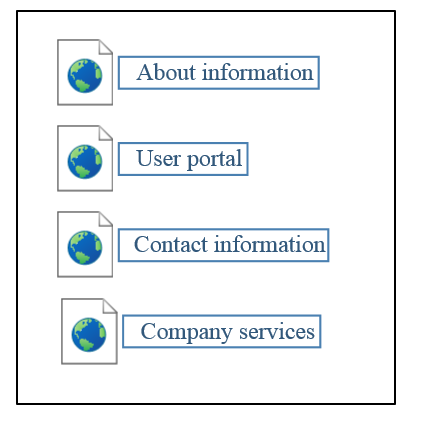
One further issue covered in a training requirement is the method of acceptance. For an airline, their senior pilots are likely to retain close links with the simulator company and go through a formal series of acceptance tests at the factory, prior to shipment and delivery of the simulator, followed by further acceptance tests following the installation.

## Analysis of existing working examples of aircraft and pilot portals

The idea of portals are to provide brief or full information of products and services that each company provides. Web applications are wide used to provide current needs. Companies sites that have been investigated:

* Panam flight academy.
* Cardif aviation training center.
* FSC training company.

In some cases portals or sites provides full stack of services like user/pilot portal with user data stored and assigned to one. Custom approach is to make a sum of all possible services in one web application: contact information, company description, company services, user portal (Fig. 1.2, Fig. 1.3, Fig. 1.4).



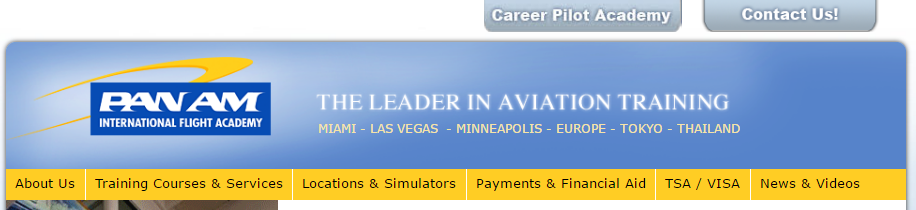
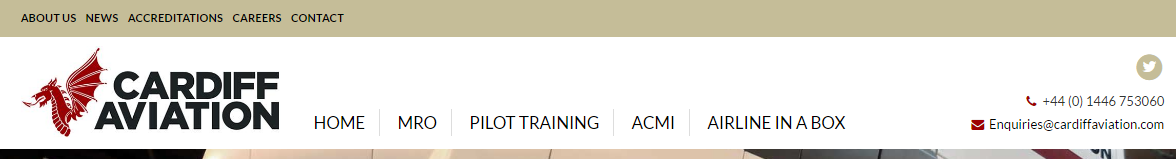


Fig. 1.4. Cardif aviation training center site.

Fig. 1.3. Panam flight academy site.

Fig. 1.2. Common company site map.

From other side, some company sites and portals provides only service for get-to-know and get familiar with products that company provides (Fig. 1.5). User cannot log in or register to become a member or get member rights on the site. The user/pilot portal is located on internal servers and provides functionality on closed basis.

Due to impossibility of getting access to closed part of user/pilot usability part, the investigation cannot be proceed and only theoretical and logical results is used.

## Client–server architecture.

Fig. 1.5. FSC site with log in functionality.

Client-server model is most used application structure these days. Due to speed of data transition nowadays, this architecture is most effective and that is why it has so much usage. The client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers, which await incoming requests.

### Client and server roles.

The client-server characteristic describes the relationship of cooperating programs in an application. The server component provides a function or service to one or many clients, which initiate requests for such services.

Servers are classified by the services they provide. For example, a web server serves web pages and a file server serves computer files. A shared resource may be any of the server computer's software and electronic components, from programs and data to processors and storage devices. The sharing of resources of a server constitutes a service.

Whether a computer is a client, a server, or both, is determined by the nature of the application that requires the service functions. For example, a single computer can run web server and file server software at the same time to serve different data to clients making different kinds of requests. Client software can also communicate with server software within the same computer. Communication between servers, such as to synchronize data, is sometimes called inter-server or server-to-server communication.

### Client and server communication.

In general, a service is an abstraction of computer resources and a client does not have to be concerned with how the server performs while fulfilling the request and delivering the response. The client only has to understand the response based on the well-known application protocol, i.e. the content and the formatting of the data for the requested service.

Clients and servers exchange messages in a request–response messaging pattern. The client sends a request, and the server returns a response. This exchange of messages is an example of inter-process communication. To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in a communications protocol. All client-server protocols operate in the application layer. The application layer protocol defines the basic patterns of the dialogue. To formalize the data exchange even further, the server may implement an application programming interface (API). The API is an abstraction layer for accessing a service. By restricting communication to a specific content format, it facilitates parsing. By abstracting access, it facilitates cross-platform data exchange.

A server may receive requests from many distinct clients in a short period. A computer can only perform a limited number of tasks at any moment, and relies on a scheduling system to prioritize incoming requests from clients to accommodate them. To prevent abuse and maximize availability, server software may limit the availability to clients. Denial of service attacks are designed to exploit a server's obligation to process requests by overloading it with excessive request rates.

## Model-view-controller design pattern.

Because of great design value Model–view–controller (MVC) became one of the most usable and popular one in development process of modern web applications. It is a software design pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. Traditionally MVC was used for desktop graphical user interfaces (GUIs), this architecture has become popular for designing web applications and even mobile, desktop and other clients.

### Description of MVC.

As with other software architectures, MVC expresses the "core of the solution" to a problem while allowing it to be adapted for each system (Fig. 1.6). Particular MVC architectures can vary significantly from the traditional description here.



Fig. 1.6. A typical collaboration of the MVC components.

The central component of MVC, the model, captures the behavior of the application in terms of its problem domain, independent of the user interface.

1. The model directly manages the data, logic, and rules of the application.
2. A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
3. The third part, the controller, accepts input and converts it to commands for the model or view.

In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them.

* + 1. A model stores data that is retrieved according to commands from the controller and displayed in the view.
    2. A view generates new output to the user based on changes in the model.
    3. A controller can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., scrolling through a document).

### Use in web applications

Although originally developed for desktop computing, MVC has been widely adopted as an architecture for World Wide Web applications in major programming languages. Several commercial and noncommercial web frameworks have been created that enforce the pattern. These software frameworks vary in their interpretations, mainly in the way that the MVC responsibilities are divided between the client and server.

Early web MVC frameworks took a thin client approach that placed almost the entire model, view and controller logic on the server. This is still reflected in popular frameworks such as PHP, Django, Rails and ASP.NET MVC. In this approach, the client sends either hyperlink requests or form input to the controller and then receives a complete and updated web page (or other document) from the view; the model exists entirely on the server. As client technologies have matured, frameworks such as AngularJS, EmberJS, JavaScriptMVC and Backbone have been created that allow the MVC components to execute partly on the client (also see Ajax).

## Analysis of examples software tools and technologies for architecture of aircraft companies sites and portals.

Due to investigation process, it has been found out that simple architectural patterns and models are commonly used. The usage of not deep models provides easiness of understanding and development process. Because of complicated business logic is redundant in portal’s functionality it is effective to use less demanded to resources models. Common use of simple architectural decisions caused by effective spent of resources that are need to create fully functional software product. For instance, portals that have been analyzes above provide only basic business logic like information display.

From other side, complicated business solutions with inner functionality like authentication and authorization need to implement more reliable and complicated models or patterns. Security is important part of these systems. The implementation of hard-maintained functions, or functionality that cannot be monitored, demands usage of solutions, methods or even external products that are costly and difficult to get to know and even use.

Complex solutions can be got, bought or even developed by developers that are creating product. Free products are wide spread in a sphere of development web applications, even provided by owner of some enterprise package for some languages. The example is Java. Oracle provides Java Enterprise. Java Enterprise Edition or Java EE is a widely used computing platform for enterprise software. The platform provides an API runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications.

### Session management.

One huge variety of functionality and possibilities that JEE provides is session management. This is mostly used for authentication on web application.

Session is a conversional state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

In Java EE HttpServelet interface is used. A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers The servlet container uses this interface to create a session between an HTTP client and an HTTP server. The session persists for a specified time period, across more than one connection or page request from the user. A session usually corresponds to one user, who may visit a site many times. The server can maintain a session in many ways such as using cookies or rewriting URLs.

This interface allows servlets to:

1. View and manipulate information about a session, such as the session identifier, creation time, and last accessed time
2. Bind objects to sessions, allowing user information to persist across multiple user connections

When an application stores an object in or removes an object from a session, the session checks whether the object implements HttpSessionBindingListener. If it does, the servlet notifies the object that it has been bound to or unbound from the session. Notifications are sent after the binding methods complete. For session that are invalidated or expire, notifications are sent after the session has been invalidated or expired.

### Java Servlet.

A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers. Such Web servlets are the Java counterpart to other dynamic Web content technologies such as PHP and ASP.NET.

Servlets are most often used to process or store a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus, "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or rewriting URLs.

To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

The Servlet API, contained in the Java package hierarchy javax.servlet, defines the expected interactions of the web container and a servlet.

A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environment. The package javax.servlet.http defines HTTP-specific subclasses of the generic servlet elements, including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application (Fig.1.7).

Servlets can be generated automatically from Java Server Pages (JSP) by the JavaServer Pages compiler. The difference between servlets and JSP is that servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML. While the direct usage of servlets to generate HTML (as shown in the example below) has become rare, the higher level MVC web framework in Java EE (JSF) still explicitly uses the servlet technology for the low level request/response handling via the FacesServlet. A somewhat older usage is to use servlets in conjunction with JSPs in a pattern called "Model 2", which is a flavor of the model–view–controller.

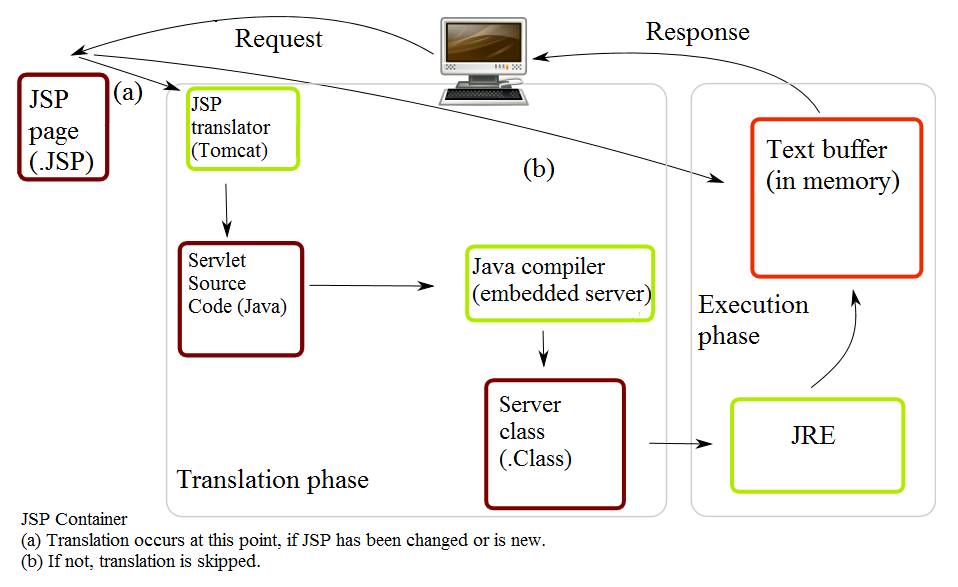
The current version of Servlet is 3.1.

Fig.1.7. Life of a JSP file.