Activity Diagrams consist of activities, states and transitions between activities and states • Activity Diagrams describe ,how activities are coordinated to provide a service ,the events needed to achieve some operation ,how the events in a single use case relate to one another ,how a collection of use cases coordinate to create a workflow for an organization.

An Activity is the process being modelled . Activities are the vertices of the diagram . An Activity is a unit of work that needs to be carried out . Any Activity takes time . An activity is like a state where the criterion for leaving the state is the completion of the activity. An Action is a step in the overall activity • The work can be documented as Actions in the activity .

Activity edges are The flow of an activity is shown using arrowed lines called edges or paths . Control-flow Transitions indicate the order of action states . Object-flow Transitions indicate that an action state inputs or outputs an object. Time could be a factor in an activity . Time events are drawn with an hourglass symbol. Conceptually, UML models information moving along an edge as a token (e.g., real data, an object or focus of control)

Activity nodes, UML 2.0 defines several types of activity nodes to model different types of information flow, Parameters nodes , Object nodes – (input or output) ,Pins - special notation for object nodes; exception pins, value pins. An initial node is the starting point for an activity .

A transition can be split into multiple paths and multiple paths combined into a single transitions by using a synchronisation bar . A synchronisation may have many in-arcs from activities and a number of out-arcs to activities. A fork is where the paths split . On an occurrence of the transition all the activities with arcs from the transition are initiated . A fork node splits the current flow through an activity into multiple concurrent flows. A join is where the paths meet .

A decision point shows where the exit transition from a state or activity may branch in alternative directions depending on a condition. A decision involves selecting one control-flow transition out of many controlflow transitions based on a condition. Each branched edge contains a guard condition. Guard expressions (inside []) label the transitions coming out of a branch. A merge point brings together alternate flows into a single output flow - note that it does not synchronise multiple concurrent flows

The Start state is the entry point to a flow. There can be several End states – multiple End states can be used to indicated different follow-on processes from a particular process. Start and End states can have actions too. Malformed diagrams – it is possible to form ill-formed diagrams that require multiple activations of activities or can allow deadlock. Swimlanes (or activity partitions) indicate where activities take place.

A diagram use case is typical interaction between a user and the system under development. It is used to capture some functionality to be provided by the software system. A use case is a usage of the system that provides an observable and (usually) meaningful result. The usecase documentation (diagrams and/or text) should delineate the series of steps that take place during the interaction and include different ways that this interaction could play out

System : sets the boundary of the system in relation to the actors who use it (outside the system) and the features it must provide (inside the system) Actor : A role played by a person, system or device that has a stake in te succesful operation of the system. Use case : Identifies a key feature of the system.

Without these features, the system will not fulfill the user/actor requirements, each use case expresses a goual the system must achieve