A dialog is the construction of interaction between two or more beings or systems. In HCI, a dialog is studied at three levels –

- **Lexical** – Shape of icons, actual keys pressed, etc., are dealt at this level.
- **Syntactic** – The order of inputs and outputs in an interaction are described at this level.
- **Semantic** – At this level, the effect of dialog on the internal application/data is taken care of.

**Dialog Representation**

To represent dialogs, we need formal techniques that serves two purposes –

- It helps in understanding the proposed design in a better way.
- It helps in analyzing dialogs to identify usability issues. E.g., Questions such as “does the design actually support undo?” can be answered.

**Introduction to Formalism**

There are many formalism techniques that we can use to signify dialogs. In this chapter, we will discuss on three of these formalism techniques, which are –

- The state transition networks \( STN \)
- The state charts
- The classical Petri nets

**State Transition Network \( STN \)**

STNs are the most spontaneous, which knows that a dialog fundamentally denotes to a progression from one state of the system to the next.

The syntax of an STN consists of the following two entities –

- **Circles** – A circle refers to a state of the system, which is branded by giving a name to the state.
- **Arcs** – The circles are connected with arcs that refers to the action/event resulting in the transition from the state where the arc initiates, to the state where it ends.

**STN Diagram**
StateCharts

StateCharts represent complex reactive systems that extend Finite State Machines (FSM), handle concurrency, and add memory to FSM. It also simplifies complex system representations. StateCharts have the following states −

- **Active state** – The present state of the underlying FSM.
- **Basic states** – These are individual states and are not composed of other states.
- **Super states** – These states are composed of other states.

**Illustration**

*For each basic state \( b \), the super state containing \( b \) is called the ancestor state. A super state is called OR super state if exactly one of its sub states is active, whenever it is active.*

Let us see the StateChart Construction of a machine that dispenses bottles on inserting coins.

The above diagram explains the entire procedure of a bottle dispensing machine. On pressing the button after inserting coin, the machine will toggle between bottle filling and dispensing modes. When a required request bottle is available, it dispense the bottle. In the background, another procedure runs where any stuck bottle will be cleared. The ‘H’ symbol in Step 4, indicates that a procedure is added to History for future access.
Petri Nets

Petri Net is a simple model of active behavior, which has four behavior elements such as — places, transitions, arcs and tokens. Petri Nets provide a graphical explanation for easy understanding.

- **Place** – This element is used to symbolize passive elements of the reactive system. A place is represented by a circle.
- **Transition** – This element is used to symbolize active elements of the reactive system. Transitions are represented by squares/rectangles.
- **Arc** – This element is used to represent causal relations. Arc is represented by arrows.
- **Token** – This element is subject to change. Tokens are represented by small filled circles.

Visual Thinking

Visual materials has assisted in the communication process since ages in form of paintings, sketches, maps, diagrams, photographs, etc. In today's world, with the invention of technology and its further growth, new potentials are offered for visual information such as thinking and reasoning. As per studies, the command of visual thinking in human-computer interaction HCI design is still not discovered completely. So, let us learn the theories that support visual thinking in sense-making activities in HCI design.

An initial terminology for talking about visual thinking was discovered that included concepts such as visual immediacy, visual impetus, visual impedance, and visual metaphors, analogies and associations, in the context of information design for the web.

As such, this design process became well suited as a logical and collaborative method during the design process. Let us discuss in brief the concepts individually.

**Visual Immediacy**

It is a reasoning process that helps in understanding of information in the visual representation. The term is chosen to highlight its time related quality, which also serves as an indicator of how well the reasoning has been facilitated by the design.

**Visual Impetus**

Visual impetus is defined as a stimulus that aims at the increase in engagement in the contextual aspects of the representation.

**Visual Impedance**

It is perceived as the opposite of visual immediacy as it is a hindrance in the design of the representation. In relation to reasoning, impedance can be expressed as a slower cognition.

**Visual Metaphors, Association, Analogy, Abduction and Blending**

- When a visual demonstration is used to understand an idea in terms of another familiar idea it is called a visual metaphor.
- Visual analogy and conceptual blending are similar to metaphors. Analogy can be defined as an implication from one particular to another. Conceptual blending can be defined as combination of elements and vital relations from varied situations.

The HCI design can be highly benefited with the use of above mentioned concepts. The concepts are pragmatic in supporting the use of visual procedures in HCI, as well as in the design processes.

**Direct Manipulation Programming**

Direct manipulation has been acclaimed as a good form of interface design, and are well received by users. Such processes use many source to get the input and finally convert them into an output as desired by the user using inbuilt tools and programs.
“Directness” has been considered as a phenomena that contributes majorly to the manipulation programming. It has the following two aspects.

- Distance
- Direct Engagement

**Distance**

Distance is an interface that decides the gulfs between a user’s goal and the level of explanation delivered by the systems, with which the user deals. These are referred to as the *Gulf of Execution* and the *Gulf of Evaluation*.

**The Gulf of Execution**

The Gulf of Execution defines the gap/gulf between a user’s goal and the device to implement that goal. One of the principal objective of Usability is to diminish this gap by removing barriers and follow steps to minimize the user’s distraction from the intended task that would prevent the flow of the work.

**The Gulf of Evaluation**

The Gulf of Evaluation is the representation of expectations that the user has interpreted from the system in a design. As per Donald Norman, *The gulf is small when the system provides information about its state in a form that is easy to get, is easy to interpret, and matches the way the person thinks of the system.*

**Direct Engagement**

It is described as a programming where the design directly takes care of the controls of the objects presented by the user and makes a system less difficult to use.

The scrutiny of the execution and evaluation process illuminates the efforts in using a system. It also gives the ways to minimize the mental effort required to use a system.

**Problems with Direct Manipulation**

- Even though the immediacy of response and the conversion of objectives to actions has made some tasks easy, all tasks should not be done easily. For example, a repetitive operation is probably best done via a script and not through immediacy.
- Direct manipulation interfaces finds it hard to manage variables, or illustration of discrete elements from a class of elements.
- Direct manipulation interfaces may not be accurate as the dependency is on the user rather than on the system.
- An important problem with direct manipulation interfaces is that it directly supports the techniques, the user thinks.

**Item Presentation Sequence**

In HCI, the presentation sequence can be planned according to the task or application requirements. The natural sequence of items in the menu should be taken care of. Main factors in presentation sequence are –

- Time
- Numeric ordering
- Physical properties

A designer must select one of the following prospects when there are no task-related arrangements –

- Alphabetic sequence of terms
• Grouping of related items
• Most frequently used items first
• Most important items first

**Menu Layout**

• Menus should be organized using task semantics.
• Broad-shallow should be preferred to narrow-deep.
• Positions should be shown by graphics, numbers or titles.
• Subtrees should use items as titles.
• Items should be grouped meaningfully.
• Items should be sequenced meaningfully.
• Brief items should be used.
• Consistent grammar, layout and technology should be used.
• Type ahead, jump ahead, or other shortcuts should be allowed.
• Jumps to previous and main menu should be allowed.
• Online help should be considered.

Guidelines for consistency should be defined for the following components –

• Titles
• Item placement
• Instructions
• Error messages
• Status reports

**Form Fill-in Dialog Boxes**

Appropriate for multiple entry of data fields –

• Complete information should be visible to the user.
• The display should resemble familiar paper forms.
• Some instructions should be given for different types of entries.

Users must be familiar with –

• Keyboards
• Use of TAB key or mouse to move the cursor
• Error correction methods
• Field-label meanings
• Permissible field contents
• Use of the ENTER and/or RETURN key.

**Form Fill-in Design Guidelines** –

• Title should be meaningful.
• Instructions should be comprehensible.
• Fields should be logically grouped and sequenced.
• The form should be visually appealing.
- Familiar field labels should be provided.
- Consistent terminology and abbreviations should be used.
- Convenient cursor movement should be available.
- Error correction for individual characters and entire field’s facility should be present.
- Error prevention.
- Error messages for unacceptable values should be populated.
- Optional fields should be clearly marked.
- Explanatory messages for fields should be available.
- Completion signal should populate.