Module 4: Process of Interaction Design, Prototyping, Construction

- ☐ Interaction Design Process
- ☐ Prototyping and Conceptual Design
- ☐ Interface Metaphors and Analogies

4.1. Introduction

Till now we have seen examples of good and bad design. We explored the importance of good interaction design. In this chapter we will take a closer look at the interaction design process. We will explore each step in detail. Further we will take an in-depth look at prototyping and conceptual design. We will also learn about interface metaphors, their powers and drawbacks.

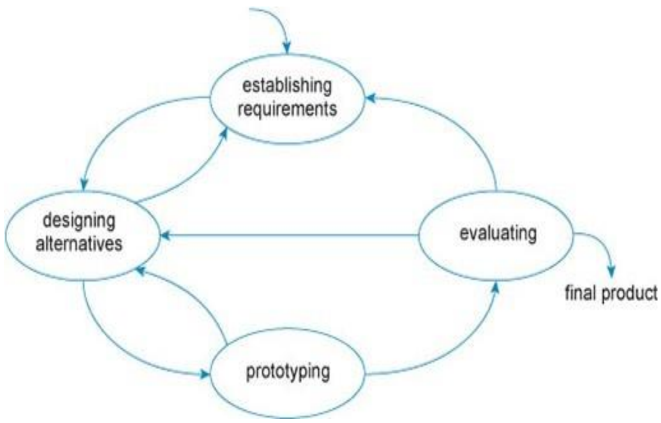
4.2. Interaction Design Process

On a daily basis we continuously interact with many different products. From physical devices like TV and fridge to websites and social media on our phones and computers. Some of the interactions are enjoyable and even fun while others are frustrating. The process of interaction design tries to ensure the usability of a product and a good user experience, The interaction design process is a user-centric approach. This means that users' perspective is considered throughout development. Having users' concerns guide the development helps to ensure a good user experience.

There are 4 stages of the interaction design process:

1. Establishing Requirements
2. Designing Alternatives
3. Prototyping
4. Evaluation

The stages of the interaction design process are illustrated in figure 4.1. It is important to note that the interaction design process is cyclic. The process is looped over multiple times until satisfactory design (final product) is reached.



4. 2.1. Establishing Requirements

The first step is to establish what is needed: The interaction design process' success (fie first dependent on successfully establishing detailed Interaction design is highly centric approach and users' needs must be understood well. a user-c

Requirements are the needs of the users and stakeholders that have to be fulfilled by the product.'

What is the current scenario? What is the user trying to accomplish? How does the user interact with the product? How does the user experience this interaction? What is the environment in which the interaction takes place? Asking such questions helps in understanding users' needs and establishing the requirements.

To successfully establish the requirements it is essential to identify the product's target audience. Additionally, modern products cater to diverse groups of users and each of their respective needs also have to be consider. For example, in a modern well-designed elevator, the buttons also have braille script embossed, thus making it easier to use for people who are visually impaired.

Furthermore, even though this is a User-centric approach other stakeholders cannot bey neglected: Their goals ' objectives and needs also have to be taken into account.

Requirement can be of two types, functional and non-functional. Needs and requirements of users and stakeholders are gathered and established through various techniques like questionnaires, interviews, demonstrations, etc. Requirement gathering techniques have been discussed earlier.

4.2.2. Design Alternatives

Design alternatives are the development of interfaces or systems that do a better job of meeting the needs of users and stakeholders/

This is the core step of the interaction design process. Here designers actually think of ways to meet the needs and requirements outlined in the previous step. It is essential/ for the designer to consider both functional and non-functional requirements: It is essential for the designer to consider many different ways to meet the needs of Users.

At every part of this process alternative options should be looked at. Even an experienced designer is virtually never able to make the best/final design in the first draft itself. Only when a large number of alternatives area considered does the best design emerge. This requires a highly creative designer.

This step can be further divided into two major parts.

1. Conceptual design

2. Physical design

Conceptual design entails the formulation of the conceptual model. A functional overview of the product is given by the conceptual model, In simple words, the conceptual model describes what the product does, how it behaves, how the user interacts with it and what it looks like, Ahead in this chapter conceptual design is discussed more in-depth.

Physical design doesn't dive into the overall flow or functionality of the product. Rather it is a detailed look into the 141 (look and feel) of the product, Physical characteristics of the product are designed. This includes selecting icons' design, menu design, transitions/ animations and colour. Since a wide variety of options are available it is important for the designer to consider multiple alternatives

To promote the consideration of alternatives brainstorming technique is often used.

Brainstorming is nicely summed up in this saying "'In brainstorming, we aim squarely at a design problem and produce an arsenal of potential solutions/'

(This is a very effective technique but only works when there is more than one designer; A design team gathers and everyone voices their ideas and opinions. All ideas are discussed and considered. Thus many different ideas get explored. Plus individual ideas are strengthened from the feedback of other designers.

4.2.3. Prototyping

Once designs are narrowed down, It is important to test the suitability of the design. This is the-purpose of a prototype. A prototype helps users and stakeholders to judge the design's functionality and usability. Since the goals to create interactive products, the, most-sensible way of evaluating such products is by using interactive versions of the design. Prototypes are such interactive versions of the design:

Real-life user interactions, made possible through prototypes, play an invaluable role in testing the suitability and validity of a design.

In the-interaction design process it is essential to create prototypes. The type and complexity of a prototype designed depends on many factors. It can be highly detailed, expensive and have a lot of the functionality of the final product or it can even be a simple paper-based representation of the idea.

4.2.4. Evaluation

Using prototypes designs/ideas are evaluated to see whether they meet the needs requirements established in the first stage of the interaction design process. Where there is scope for improvement is also determined.

Evaluation is a set of techniques used to determine whether the needs of the user and stakeholders are being met.

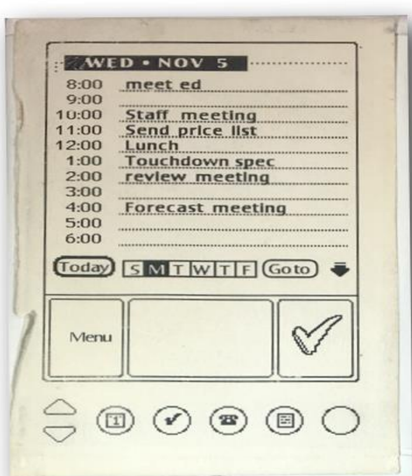
Insights and feedback gathered inform the next design iteration. Usually, after a prototype is evaluated, it is redesigned, prototyped, evaluated, redesigned, prototyped, evaluated so on.. It is important to note that the interaction design process is cyclic and it is very common for a product to go through multiple such iterations till final design is reached, Additionally, during evaluation new requirements may also surface.

4.3. Prototyping and Conceptual Design

4.3.1. Prototyping

A prototype is an interactive representation of the design that allows users and stakeholders to judge its suitability/ Since the goal is to create interactive products, so isn't interacting with them the most sensible way of evaluating such products? Prototypes are such interactive versions of the design

A prototype which is nothing but a simple drawing; source: Uxpin



The prototypes can be very rudimentary. A fascinating example is of the founder of Palm Pilot He cut a block of wood to the approximate shape and size of his design vision, illustrated in figure 4.7. Then for several weeks he carried it around in his pocket and pretended to use it when the need arose. He even took pretend notes in meetings. By changing the paper glued to the face of the block he also tried out different button configurations:

4.3.1.1 Why make Prototypes?

Human interactions are inherently complex. No matter how experienced the designer is it is almost impossible to make a perfect first design. Real-life user interactions, made possible through prototypes, play an invaluable role in getting relevant feedback. A lot of insights can be gained by observing the reaction and interaction of users with the prototypes.

It is often said, "Users can't tell you what they want, but when they see something and get to use it, they soon know what they don't want".

Prototypes play an essential role in the demonstration and evaluation of designs/ideas: Designs/ideas are evaluated to see whether they meet the user requirements and identify where there is scope for improvement, 'Insights and feedback gathered are inputs in creating the next design iteration,

Usually, after a prototype is evaluated, it is redesigned, prototyped, evaluated, redesigned, prototyped, evaluated so on... Most designs go through multiple such iterations.

4.3.1.2 Types of Prototyping

The type and complexity of a prototype created is dependent on the purpose it is meant to serve. Type is also influenced by the stage of development.

During earlier iterations and for simple idea demonstrations we often see Mow-fidelity prototypes being used, For later iterations, detailed testing and/or for marketing we often see high-fidelity prototypes being used We will now see what low-fidelity and high-fidelity prototypes are.

4.3.1.2.1 Low-Fidelity Prototyping

Low fidelity prototype often does not closely resemble the final product in either look or functionality, It is often a very bare-bones representation pf the design like in figure 4.2.4. They are often made up of relatively inexpensive materials like paper and cardboard.

Often they do not perform the functionality, they only represent it. The benefit is that low fidelity prototypes can be made very quickly and very easily. It is also usually very cheap to produce. That also makes them very simple, cheap and fast to modify. This makes them particularly useful during the exploration of alternative design ideas. These fast, flexible and inexpensive prototypes encourage exploration of different designs and ideas: Note low- fidelity prototypes are almost never integrated into the final product!

Some types of low-fidelity prototypes are discussed below:

1. Storyboarding

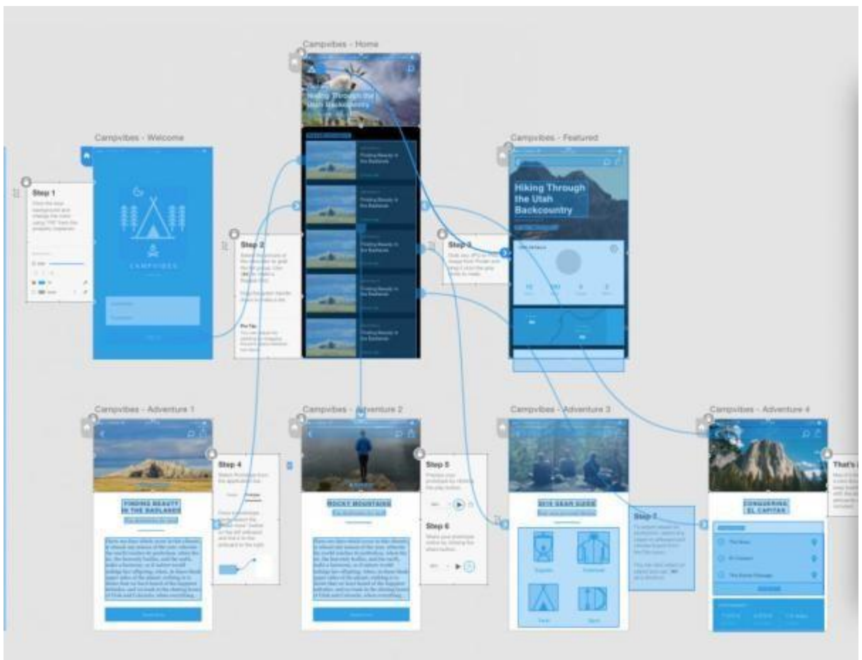
It can be a series of text/series/scenes depicting the use and interaction of a product: Often a certain scenario will be depicted in storyboarding. Refer figure 4.9. it gives the stakeholders a chance to see and understand the flow of events and the interactions with products that can take place in a certain scenario.

2. Prototyping with index cards

Index cards are used. Index cards are 3x5 inch pieces of cardboard. This IS often used while developing websites or phone apps. Each index card will represent a screen, the flow of the app or website can be simulated. It is simple and gives a good understanding to users and stakeholders about the functioning and flow of the product, Example in figure

3. Wireframes

It is often used for websites and mobile apps. It gives the structural design overview of their system. Wireframes help easily demonstrate the functionality and flow of the app or website. In many ways it is similar to prototyping with index cards' However unlike prototyping with index cards, wireframes are designed in software and thus more interactive. It is also important to note that flow shown in wireframe is meant to resemble the flow of the product but the look and feel shown is only for demonstrative purposes and not meant to resemble the look and feel of the actual product.



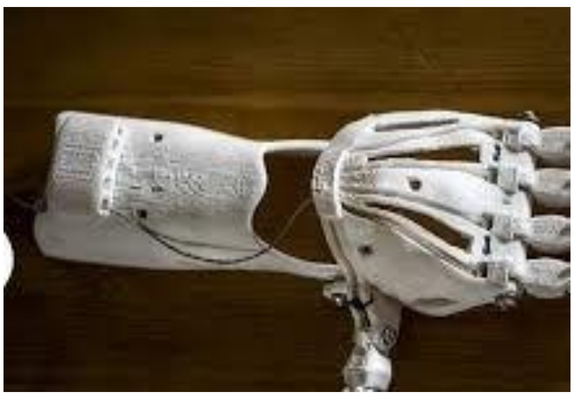
4.3.1.2.2 High-Fidelity Prototyping

Fidelity is defined as "the degree of exactness with which something is copied or reproduced". Thus as the name suggests, a high-fidelity prototype more closely resembles the final product. It also provides much more functionality of the product than a low- fidelity prototype. Therefore, more realistic -interactions occur between user and product. Interactions more closely resemble that of the final product/ Furthermore, since often many of the functionalities of the final product are present in these prototypes, they can be used to test out some of the technical issues. The drawback of high fidelity prototypes is that they are often more expensive and time-consuming to build.

High-fidelity prototypes are also often used in the marketing of the idea/product. Startups are often seen pitching their ideas to investors using high fidelity prototypes.

Both software and hardware high fidelity prototypes are often made by the tinkering and modification of existing products. This method is widely used due to it being a cost-efficient and quick way of building the prototype

Banzi (2009) comments that: "Getting cheap toys or old discarded equipment and hacking them to make them do something new is one of the best ways to get great results." However, with the reduction of cost of 3D scanning and printing, it is now more common to see accurate, to-scale 3D printed prototypes of designs/ideas.



For software products, there are many tools to help make a high fidelity prototype. They are very useful because they can very closely simulate some of the product's missing Tools like Adobe XD.

4.3.1.3 Three Main Approaches to Prototyping

1. Throw-away

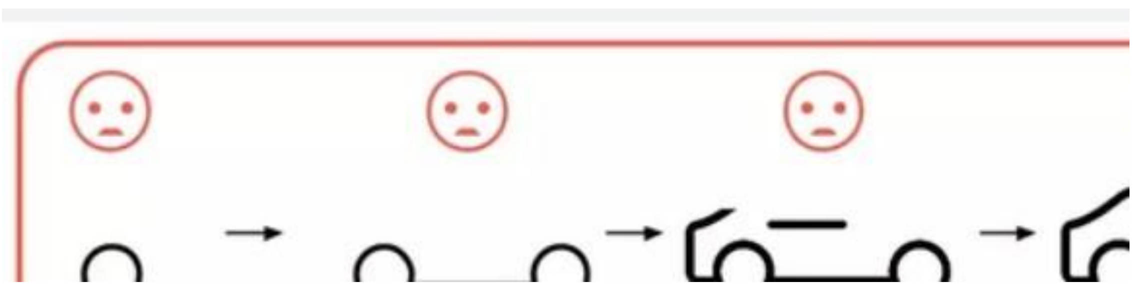
The prototype is built. Used for testing and insights are gained. These insights help inform the design of the final product. Then as the name suggests the actual prototype itself is thrown/discarded. Throwaway prototypes are useful for exploring ideas and getting feedback from end-user?.

They are also often used to answer questions and once the questions have been answered they are 'thrown away'. They are never included in the final product.

Most low fidelity prototypes are examples of throw-away prototyping. For example, while designing a new phone, a to-scale cardboard cutout of the phone is made, to see how well it fits in the hand. Will users find the proposed size too big? After testing and getting feedback, the question is answered and the prototype is thrown away.

2. Incremental

The entire product is broken down into independent components. Each 'increment' is when a new component is ready and integrated. The product is ready when all components have been completed. The process of incremental prototyping is illustrated below.



The benefit is that users are able to provide feedback on the ready components, while the others are still under development. Often this feedback also helps in improving the components that are still under development for example, a new spreadsheet processor (similar to excel) where users can open and save spreadsheets, but users cannot filter, sort, insert graphs, use sum and count functions because they are still under development.

3. Evolutionary

In this technique, the prototype is not discarded. It forms the base from which the ne4, iteration is designed The process of evolutionary prototyping illustrated in figure.

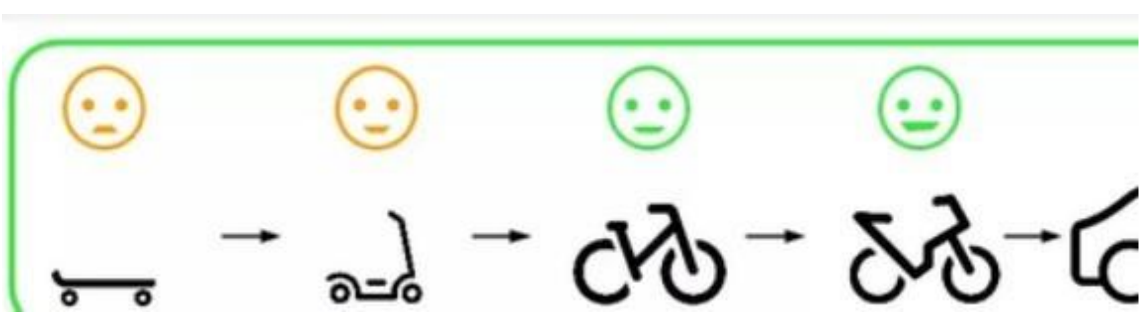


Figure 4.17: Illustration ofthe process ofevolutionary prototyping; source: Quora

The idea is that the prototype should go through multiple refinements until the' prototype itself becomes the solution:

This is especially useful when some of the requirements are vague and unclear. Users provide feedback at the end of each iteration to help make the prototype closer to the solution in the next iteration.

4.3.2. Conceptual Design

The main activity of conceptual design is to turn the needs and requirements gathered into a conceptual model. The conceptual model describes what the product does, how it behaves, how the user interacts with it, what it looks like and will the user perceive the product as intended.

4.3.2.1 Guiding Principles of Conceptual Design

There are four principles that guide the conceptual design. During the entire conceptual design process, designers should always keep these principles in mind.

1. Keep an open mind but never forget the users and their context

To turn the requirements gathered into a conceptual model, different ways to fulfill these requirements are considered. It is essential to keep an open mind to the various different ways/techniques/ideas in which the requirements can be fulfilled. Since only by considering many different alternatives, it will lead to the best possible design.

Additionally, at all times the user's perspective should be kept in mind. Designers should continuously try to see things from the user's perspective. This will ensure the meeting of users' needs and a good user experience. This is extremely important. Example of this is that many car manufacturers use an empathy suit. This suit restricts the movement of the wearer. Thus this suit helps young designers to empathize with older users who often have some loss of mobility. Ford the inventor of such a suit and calls it 'Third Age Suit'.

2. Discuss ideas with other stakeholders as much as possible

Not only the users but it is also essential not to neglect the other stakeholders' needs and concerns. Additionally continuous discussions with the other stakeholders provide instant feedback to designers. Sometimes designers may misunderstand some of the requirements and the discussion and feedback helps to ensure that the designers are on the right track

3. Use prototyping to get rapid feedback

As we have discussed earlier in the chapter, prototyping is invaluable to providing reliable and accurate feedback to designers. Best way to demonstrate ideas is with interactive representations of them.

We often see low fidelity prototypes being used to continuously demonstrate ideas to stakeholders. Since low fidelity prototypes are cheap and incredibly fast to build therefore designers can get rapid feedback from stakeholders.

4. Iterate, iterate and iterate

Even for experienced designers it is almost impossible for the first time. Improving the design/ model. Conceptual model continuously evolves, with each iteration feedback is gathered and the conceptual model is improved, till a satisfactory conceptual model is reached

4.3.2.2 Developing the Conceptual Model

The needs and requirements gathered forms the basis from which the conceptual model is made. A deep dive into the data gathered guides the development of the conceptual model. From the data designers also extract the user experience goals. During development it is essential to keep the users' perspective and the user experience goals in mind.

In the section we discuss the approach to developing the conceptual model.

Developers have to consider the following questions while developing the conceptual model. Which metaphors would help users to understand the product? Which interaction type/types would best assist the activities performed by users? Do using different interface types provide alternative design insights? Now we will look at each of these factors.

4.3.2.2.1 Interface metaphors

Using a good metaphor makes it easy to understand and pick up the software's functionality. Metaphors is this use of real-world scenarios that people are familiar with to represent and make easier the understanding of new interface concepts and designs.

For example, when word processors were initially released, they used the typewriter metaphor. They designed the keyboard of a computer to be very similar to a typewriter• Shown in figure 4.20 (a) and (b). Thus it was a powerful and intuitive metaphor. Experienced typewriter typists were able to predict the behavior keyboard due to this metaphor.

A good metaphor is invaluable in increasing the initial familiarity of the user with the product. It even may be very enjoyable and memorable at the start. Ahead in this chapter we discuss metaphors in greater detail.

4.3.2.2.2 Interaction types

Oxford Dictionary defines interaction as "an occasion when two or more people or things communicate with or react to each other". We are concerned with the interaction of the proposed system/product and the user.

Interaction types are different ways in which the product and the user react to each other. The interaction of users with the system heavily influences the overall user experience. Choosing the right interaction type or combination of interaction types is essential to ensure functionality and usability of the product. The type or combination of types to be used is dependent on the scenario, functionality and the intended target audience. There is no one right type or combination of types which is correct for all scenarios. A combination of multiple interaction types can be seen in most conceptual models.

Interaction can broadly be categorized into 4 types:

1. Instructing
2. Conversing
3. Manipulating
4. Exploring

Now let us see each category in more detail.

1. Instructing

As the name suggests this interaction involves the user giving the system command(s). 'Instruction' is issued to the system by the user. The 'instruction' can be issued in a number of ways. Clicking on a menu option, typing commands, using function keys, etc are all examples of this type of interaction. 'Instructing' is always a one way conversation between user and system.

Almost everyone who has ever used a phone or a computer has had this type of interaction. Examples include, instructing windows to start the task manager by pressing control alt delete, hitting control + s to save a file, illustrated in figure 4.21 clicking on a contact to initiate a call on the phone, illustrated in figure 4.21 b

2. Conversing

This interaction involves a two-way conversation between user and system. The system tries to mimic real life conversations. The system acts more like a human providing a reply. The user asks something and the system provides a reply.

Nowadays using speech to text and text to speech software, systems can mimic an actual conversation. Users may physically voice questions and the systems can physically voice replies. Google assistant, Siri, Alexa are examples of such systems.

Simplest example of this is When you google something, you are asking a question to Google and the search results is Google's reply. Another example is asking Google assistant, Siri or Alexa what is the time?

converging is not only limited to voice assistants. Nowadays complicated natural language processing (NLP) systems also can be seen frequently. Many banking, ticket booking, e-commerce websites will often have an automated chatbot. Where you can ask the chatbot questions and based on keywords in your question it will provide a response.

The drawback is often such interactions can become cumbersome and inconvenient. One such example of this is the automated voice heard while calling most banks' customer support. An automated voice provides many options and the user has to choose. Example 1 - for credit card related queries, 2 - checking the balance in your account, 3 - for reporting fraudulent transactions. Uses and has to choose on option and type in the corresponding option number. Often there are multiple menu layers and this process is repeated many times. This type of interaction often becomes extremely frustrating.

3. Manipulating

Manipulate is defined by the Oxford dictionary as "to control something using the hands, This is the essence of this type of interaction. This interaction involves the 'manipulation, of virtual or physical objects.

Virtual objects are manipulated in a similar way to in the physical world. This type of interaction can be quite intuitive because the user automatically relates to his/her physical world experience. Metaphors are often heavily incorporated in this type of interaction, thus making understanding and learning extremely intuitive. Metaphors are explored in greater detail later in this chapter.

Selecting, dragging, opening, closing, zooming in and out using touch gestures on a smartphone are all examples of this type of interaction.

Examples include dragging and dropping a file in windows, zooming in and out on a phone, selecting some text in word using the mouse.

It can also involve use of physical controllers and sensors. An example is using a controller to control the movements on a screen avatar on the gaming console Wii.

Advantage is that the basic functionality is learnt very quickly. Error messages are rarely needed.

Users get immediate feedback on their actions. Example on a smartphone using the gesture, a user is trying to enlarge a photo. If this does not work the user immediately comes to know because there will be no movement on the screen. Additionally, an error message is not needed, the user has realized the gesture has not worked.

4. Exploring

AS the name suggests this interaction involves the travel through (exploration) of a virtual or physical environment. VR is an example of a virtual environment. Physical space with sensors embedded, like smart homes, are examples of physical environments. Most common occurrence of this type of interaction can be seen in 3D games like Minecraft and Pubg, seen in figure 4.25 and figure 4.26. They involve exploring a 3D virtual environment. There are a lot of other types of interactions that also occur throughout the name, but a substantial part of the game is the exploration of a virtual environment.

In the last few years there has been a substantial increase in the use of virtual reality (VR) and augmented reality (AR). Google daydream allows users to explore some cities through VR, shown in figure 4.27.

4.3.2.2.3 Interface types

We have just seen the different types of interactions that can take place between system and user. Conceptual design also includes deciding the medium through which the above discussed interaction occurs.

Interface is the medium through which the interaction occurs between user and system.

There are many different types of interfaces like command-based, virtual reality, web, mobile, speech, pen, touch, air-based gesture, shareable, tangible, augmented and mixed reality. We have already discussed them in detail in chapter 2.

Different products and different scenarios require different interfaces. Additionally, a product is not limited to a single interface. In a conceptual model it is important not to only consider a single interface type. Different interface types encourage and support different perspectives of the system/product. Therefore considering multiple different interfaces prompts the consideration of alternatives.

4.3.2.2 Expanding the Conceptual Model

In this section the conceptual model ideas that were produced in the previous section are looked at in greater detail and expanded on. It is important to scrutinize each idea in much greater detail before they are prototyped and tested.

Designers accomplish this by ask the following 3 questions:

1. **What functions will the product perform?**

Establishing functions/tasks supported by the system is a fundamental part of conceptual design. Furthermore, it is important to understand and establish which tasks are to be performed by the system and which tasks are to be performed by the user.

For example in a travel agency website, the task of providing holiday recommendations is the responsibility of the product but the final selection of holiday for booking is the responsibility of the user.

2. **How are the functions related to each other?**

Functions/tasks could be related and connected to each other. It is important to understand and establish any such relations/connections. This is essential since often one task can affect another. Also the relation between tasks may put constraints on the execution. For example in a travel agency website, the task of making the hotel reservations can only be done once the user has selected and booked a vacation.

3. **What information is needed?**

Often some data/information is required to perform a certain task. It is essential to design the conceptual model in a way that provides/retrieves the data when needed. It is also essential to establish what information is being passed between the user and system. How the information is presented. Additionally, structure of data can cause issues and has to be kept in mind.

For example in a travel agency website, international travel's hotel reservations may require the passport details of the traveler. Thus the hotel reservation task is dependent on having the passport details data.

4.4 Interface Metaphors and Analogies

4.4.1. What are interface Metaphors and Analogies

When the LOGO language had been developed, to teach children, the metaphor of a turtle dragging its tail in the sand was used. Therefore, it was very easy for children to understand this real-world situation and they were easily able to grasp how to draw pictures in LOGO language.

Metaphors or analogies is this use of real-world scenarios that people are familiar with to represent and make easier the understanding of new interface concepts and designs.

Metaphors range from the scissors button, indicating Cut, to the folder and file cabinet representation of windows file systems.

It is dependent on the real-world connection users make between the interface's visual clue and its function. We see a lot of these examples on a daily basis throughout computer and software interfaces.

For example, on windows to delete any file, the file can be dragged to the recycle bin icon on the desktop. Recycle bin icon in figure 4.30. Users relate this action of moving files to the recycle bin to the real-world scenarios where we throw waste into the recycle/trash bin. The computer does not actually move the file to the recycle bin, it deletes it, yet it is very easy to understand what is the purpose and functionality of the recycle bin. This is the power of interface metaphors and analogies.

Another example is seen very often on e-commerce sites like Amazon, Flipkart, Walmart etc. Most of these ecommerce sites use 'add to shopping cart/trolley/basket' followed by 'checkout'. Isn't this a good metaphor and doesn't it have Similarities to the purchase process in Physical stores. Doesn't this make a purchase intuitive for first-time users?

4.4.2. Drawbacks of Interface Metaphors

It has been seen throughout the history of computing that metaphors, or analogies, have been very successful in introducing beginners to new interaction techniques. However, metaphors also have some drawbacks.

4.4.2.1 Navigation Overhead After Familiarity

Relying on a metaphor makes it easy to understand and pick up the software's functionality. It even may be very enjoyable and memorable at the start. No one really denies the value of a good metaphor for increasing the initial familiarity of the user with the product. But once the functionality is understood, the metaphor can add a significant navigation overhead.

It is often said, "The danger of a metaphor is usually realized after the initial honeymoon period".

For example, let's once again consider the windows recycle bin. The recycle bin metaphor is a very powerful metaphor. Even for a beginner, it is very easy to understand and learn how to delete files using the recycle bin. But consider if the only way to delete a file was to drag and drop it to the recycle bin. Meaning other applications may have to be minimized, file explorer may have to be repositioned and then the file has to manually dragged and dropped. Isn't this inefficient? Once the user is familiar and no longer a beginner, wouldn't it be much faster to hit the delete button on the keyboard or right-clicking and selecting delete?

4.4.2.2 Cultural Bias

In modern times software has a global reach and is not limited to either one country or one culture. At the core, metaphors rely on the users' understanding of the imagery. Additionally, users should make the connection in their mind in the same way as the designer intended. However, users are from many different cultures often with different religions, languages, norms, etc. Therefore, different cultures can often have a different interpretation of the metaphor.

It often happens that the imagery and metaphor will be perfectly understood by a country or a community but will make no sense to another.

For example,. the previously mentioned windows recycle bin metaphor is appropriate in countries like the U.S. and Canada where trash is collected in bins with the recycle symbol on it and/or the phrase commonly used is "Recycle bin". However, many countries do not collect waste in such bins and do not use the phrase 'Recycle Bin'. To such users the phrase 'recycle bin' will be meaningless. Furthermore, such Users may get confused by the recycling symbol on the recycle bin icon. The functionality of windows recycle bin is the deletion of files. But such users may think - "Are the files going to be recycled?" This metaphor works well for certain countries but not for others.

To those users who have not understood the metaphor, it will only add to the confusion. It is very hard to find a metaphor which has global recognition.

Thus, unfortunately, metaphors are extremely vulnerable to cultural bias.

4.4.2.3 Understanding/Intuition is Subjective

A metaphor in the context of interaction design means a visual metaphor that signals some functionality. It is then presumed that the user understands the metaphor in the same way as the designer intended and then successfully intuits the functionality.

However, intuition is a very subjective process, the way people make connections is unique to them and heavily influenced by their own past experiences. Users are from many different cultures, religions, languages, socio-economic backgrounds and have different past experiences. Therefore, everyone's interpretation of the metaphor is likely to be different.

Thus even within the same cultural group users may infer metaphors in different ways and significant misunderstandings can occur.

As discussed earlier to those users who have misinterpreted or have not understood, a meaningless metaphor will only add to the confusion. For example, some users may assume the scissors icon, when applied on a photo in Microsoft Word is for cropping a photo and not for 'Cut' operation.

4.4.2.4 Limits Understanding of Full Functionality

Often metaphors limit the full understanding of the functionality of the system. This issue is seen time and again. Often metaphors are very good in providing preliminary understanding. However, they can become an obstacle in the full understanding of the functionality.

For example, the Windows file system uses the filing cabinet metaphor. This metaphor for organizing documents in windows is quite easy to understand. However, there are many inherent limitations to real-world filing cabinets which are not applicable to windows file systems. For example, in the real world, you cannot have 50 folders nested inside one another. You cannot have the same file in multiple places in a filing cabinet. You cannot have a shortcut to another location of the filing cabinet. The filing cabinet metaphor is very useful for a preliminary understanding, but the analogy falls short in giving a full understanding of how the windows file system works. In fact, since the user was continuously trying to relate to their experience of a filing cabinet, the metaphor becomes an obstacle in the user's full understanding of the Windows file system.

4.4.2.5 Physical World Limitation

It is easy to find metaphors in the physical world for objects like a printer. But it can be difficult to find metaphors for processes, services and relations.

For example, it can be difficult to find physical world metaphors for changing a video's resolution, changing photos border and shading, word count, performing Statistical operations. Yet such operations are the most frequently used.

Additionally, metaphors don't scale well. In today's day and age, software is continuously Changing and evolving. Thus, a metaphor that worked nicely for a process can fail once the process grows in complexity.