1. **Compare and contrast analytical evaluation methods and empirical evaluation methods illustrating your answer with examples of each.**

**Answer:** Both analytical evaluation and empirical evaluation methods have an end objective to improve the usability performance of the product the usability goals here are effectiveness, efficiency, safety, utility, learnability, memorability. Both methods help the evaluation by making the user experience satisfying, pleasurable, rewarding, fun, provocative. The key aim of both methods is to feedback on design directions and ideas. They also help n discovering major issues which are major part of any product (application) development. Both help to resolve disagreements.

The key difference is analytical evaluation does not involve users while empirical evaluation involves users. Analytical evaluation has experts analyse the prototype in a detailed way to understand how users will understand it. On the other hand, empirical evaluations have real users (persona fit audience) to test usability metrics. The analytical evaluation method requires a prototype or fairly detailed description, it also requires an indication of who the users are and their assumed knowledge. While the empirical evaluation method can be earlier in the design process, typically in the lab for usability evaluation or later in the design process, out in the real world, to test user experience metrics for field studies evaluation.

Both of these methods make you consider the project where you have to define what are the usability goals, what are the user experience goals and how to define and operationalize these goals.

1. **A gardening app is being developed for older people that are intended to be very easy to use. Houde and Hill’s (1997) model of prototyping classifies prototypes as ‘role’, ‘implementation’ and ‘look and feel’. Describe each of: a role prototype, an implementation prototype and a look and feel prototype for the proposed app. Justify in each case how they might be used and how they fit the given category.**

**Answer:** Role Prototype: Role Prototypes are the ones which are constructed basically  
to research questions of what an artefact may do for a user. They describe the capability  
that a user would possibly advantage from and how the artefact fits in the user's life.

For the gardening app that is developed for old people and is intended to be very easy to use, a small video is created showing the app being used for the gardening purpose, by elderly people in day-to-day life. This video will show the potential user what changes could occur in the life of the user because of the app.

A prototype is handed over to some of the volunteering elderly people, these people then use (mimic) the app prototype to perform an action. These actions are then observed by the potential users and the aim to get the assumed value and practical use of the app in daily life.

Implementation Prototype: Implementation Prototypes are built primarily to answer technical questions about how a future artefact might be made to work. Designers make implementation prototypes as experiments for themselves and the design team, to demonstrate to their organization the technical feasibility of the artefact, and to get feedback from users on performance issues.

This gardening app is developed with objectives to give best Tips-Ideas-Videos about gardening in the simplest way possible. Hence the app will rely on the being able to display tips and ideas in best interactive way without any lag or issues. The app will able play videos and should be able to be operate smoothly on basic smartphones, as elderly people are more likely to use basic smart phones.

The prototype is created without having to define its look and feel or the role it will play for a user, this prototype will have basic app layout with the section to display ideas and tips for gardening, it will also have an additional window which will play the videos.

This prototype will help the designers and the development team to figure out the technical feasibility and the feedback from users on performance issues.

Look and Feel Prototype: This is the most common type of prototype, something you can touch and feel that explores what the users experience will be with an object or service.

As this gardening app is developed for elderly people, the first thing to do would be create photoshop versions of the screen to determine the fonts, font size, colour scheme and colour intensity. Also, basic versions of basic window layout and size will be developed.

A prototype will be then handed over to the users, which will be a post-it note prototype containing series of screen an older person will go through while using the app. This prototype will be used to understand the actual interaction design, the button they must press, the visibility of the fonts and colour, the window layout use and the feedback they will receive from the device as they do different task.

A hi-fidelity prototype is created to see if the older person is actually be able to see the tips & ideas and play the videos.

1. **Describe** **the characteristics that Shneiderman highlights as the distinguishing features of direct manipulation interfaces.**

**Answer:** The characteristics that Shneiderman highlights as the distinguishing features of direct manipulation interfaces are:

**Continuous representation of the object of interest:**

The users of the application can see the visual representations of the objects that they can interact with. The performed action has an immediate feedback. Example is when a task is performed in Microsoft Paint, when a pen that draws straight line chosen with drag and dropped on the drawing area, it draws a line and you can changed the length & size of the line with direct manipulation. The results can be seen immediately.

**Physical actions or labelled button press instead of complex syntax:**

Physical actions are generally more natural to humans than writing in syntax, these actions could be mouse clicks, touch gestures, pressing buttons etc. In the above example of using pen for drawing straight line can be directly related to someone using pen to draw straight line in real word. Hence these action is more natural and can be performed without any exceptional expertise. On the other hand.

**Rapid incremental reversible operations whose impact on the object of interest is immediately visible.**

Direct manipulation offers visibility of the system state hence it is easy to see if the action has outcome in right result. On the other hand, command line syntax needs experience and requires knowing the correct syntax. On top of that if the command contains a typo, it takes close observation and time to understand where is the mistake.

In Direct manipulation user can see the mistakes immediately and are able to undo it easily.

1. **Look at Fig. 1 below. Explain Hutchins et al. (1985)'s concepts of semantic distance and articulatory distance, and the distinction between them. Illustrate your answer with an example.**

Answer: Semantic Distance: Semantic distance concerns the relation of the meaning of an expression in the interface language to what the user wants to say. Evaluation is the activity that spans semantic distance in the gulf of evaluation.

The two important questions about sematic distance are:

1. *Is it possible to say what one wants to say in this language?*
2. Can the things of interest be said concisely?

For Example:

Goals Meaning of Expression

Semantic Distance

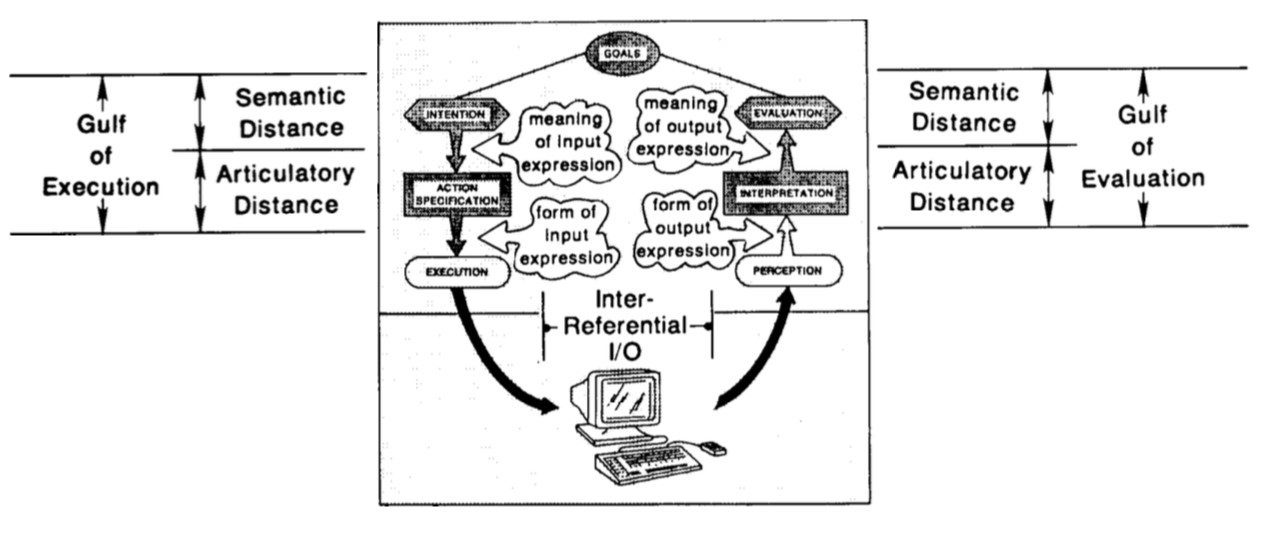
Articulatory Distance: Articulatory distance is about the relationship between the meanings of expressions and their physical form. Interpretation is the activity that spans articulatory distance in the gulf of evaluation.

Meaning of Expression

Articulatory Distance

Form of Expression

In the example



# References

Edwin L. Hutchins, J. D. (1985). *HUMAN-COMPUTER INTERACTION, 1985, Volume 1,.* Lawrence Erlbaum Associates, Inc. .

Hil, S. H. (n.d.). *What do Prototypes Prototype?* Cupertino, CA, USA: Apple Computer, Inc.

Shneiderman, B. (n.d.). *Direct Manipulation: A Step Beyond Programming Languages.*