Create UI and how to exactly describe them in either code / XML,

⇒ Declarative system

Based on our discussion so far, Android framework/system does not have an idea about what the user is doing or the app's logic.

Possible that the

Does it need to know?

requires Semantic information

⇒ Additional Semantic analysis; used for testing and debugging

⇓

"true meaning"

⇒ Accessibility Service provided by Android Framework

⇒ Goal is to make smartphones

accessible or usable by {different types of users} ⇒ visually challenged.

Android ⇒ UI Tree (discussed)

Semantics Tree

Whatever node is present in the UI Tree also can optionally add nodes to the Semantics tree.

⇓

State of the UI elements.

Select Button ⇒ { State Description ⇒ "Is it enabled or disabled"?

Toggled =

String ⇒ "Why is this button present"?

⇓

Optionally provide information to the Android system about why it is present.

All the above info is present in the APK, but the challenge it is in a form difficult to understand.

Tool ⇒ Layout Inspector (part of Android Studio)

Click on each UI element and it is possible to find semantic information corresponding to it.

1) Not all modes of communication are equivalent

(a) ⇒ Show a lot of text within a Textbox (Potentially use lazy modifiers to handle)

⇒ AccessibilityService's Talkback would try to read the entire content.

⇒ Provide a modifier called heading().

If the textual content is large, it reads only the headings.

Does the above solution solve the problem?

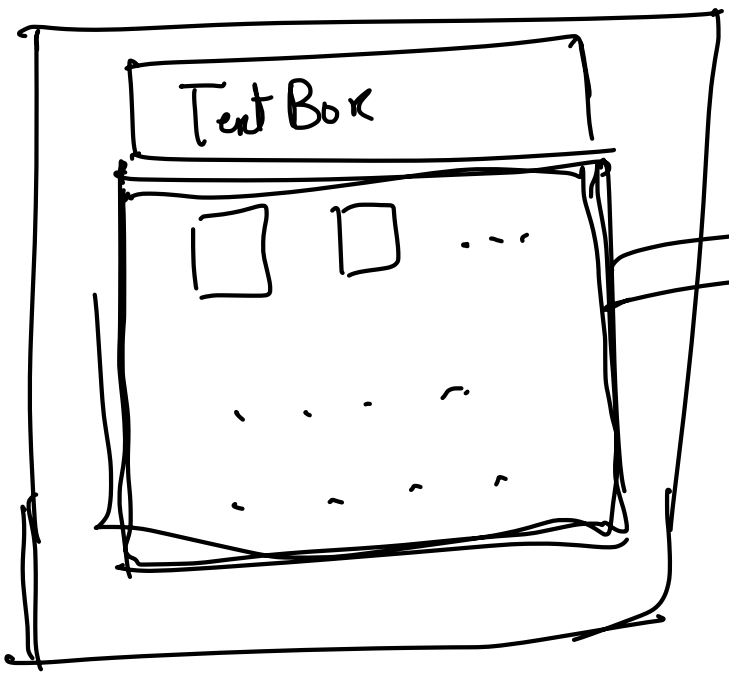
⇒ Solves it only partially ⇒ Developers need to identify what is important

(b) ⇒ What if there are images or videos?

Image (contentDescription = stringResource ("This image shows the current state"))

2) Semantic trees are often very cluttered and have deep hierarchies, making them hard to navigate.

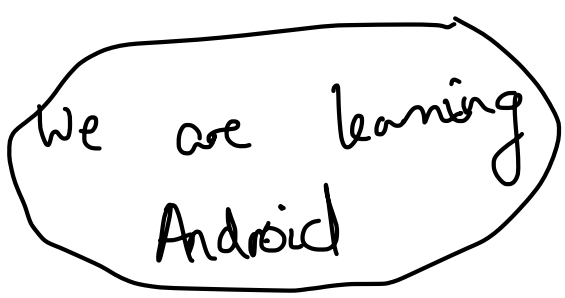
Semantic nodes can be merged with their descendents. "Merge with descendents" imply that the descendent's children becomes the node's children, and the descendent's semantics becomes the node's semantics.



3) If a custom-defined UI element is created, then its Semantics has to be defined separately using modifiers.

---

1 (c)



MC course starts at 3pm.

Traversal Group

Modifier.semantics { isTraversalGroup = true }

Indicates that the content is part of a single group and should not be split. Then TalkBack service will be able to figure out how to read out.

---

§ Is it possible that we want to separate out the semantics of the UI elements?

⇒ When the state associated with an UI element is complex, then we may prefer to create a new node.

What is the equivalent of this in traditional programming?

⇒ Creating subclasses and other ways of organizing data.

Semantics are nothing but organization of the UI state; but in a more understable form.

Is it possible that the semantics create additional performance bottlenecks?

⇒ Changing of semantic info is equivalent to recomposition of UI,

⇒ Avoid changing the semantics unless it is necessary.