

Master International, Interaction & HCID - Université Paris-Sud
Fundamentals of Human-Computer Interaction
Exam - 14 November 2018 - 3h

Authorized document: a single A4 handwritten sheet.
 Please read the questions carefully. Answer clearly, precisely and concisely.

A. Conceptual modeling (6 points)

You are asked to design an interface for a simple photo application running on a multitouch tablet. This interface must allow the user to:

- Take pictures;
- Create photo albums;
- Add, remove, change the order of pictures in an album;
- Navigate all pictures or those in an album;
- Apply filters to pictures;
- Send individual pictures or an entire album to a contact.

1. Identify the objects of interest and the operations of the conceptual model.
2. Sketch the layout of your interface and explain the general design of the interface. **Justify your choices with the rules and principles seen in class.**
3. Fill out the objects and operations tables describing the conceptual model of the complete interface. For each operation, design at least one interactions to trigger the operation. To get you started, the tables below include one object and one operation (you can change those in your design).
4. Is your interface instrumental? Explain.
 If it is not, how would you apply the design principles of instrumental interaction to improve it?

Objects table:

Objects	Representations	Properties	Operations
Picture	Thumbnail	Image, date taken	Open
...			

Operations table:

Operations	Commands	Feedback	Responses
Open a picture	Double tap the thumbnail	The thumbnail grows into a full-screen image	The picture is displayed in full screen
...			

B. Modeling interaction: spring-loaded folders (7 points)

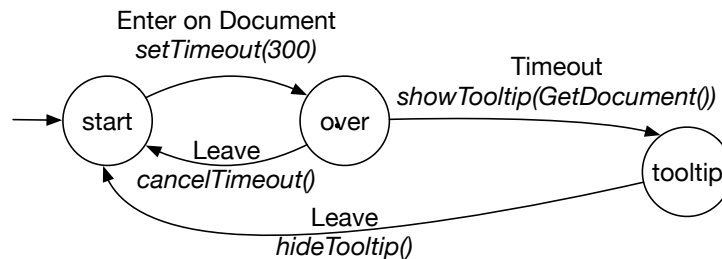
Moving files in the file hierarchy is a common operation on a desktop computer. This exercise uses state machines to describe the "spring-loaded folders" interaction technique that is available on Macintosh computers.

The following events and actions can be used in the state machines :

- *Down*, *Up* and *Move* are events sent by the mouse;
- *Enter* and *Leave* are events sent when the cursor enters or leaves an objects of type *Document*, *Folder* or *Window*, corresponding to document icons, folder icons and windows;
- *TimeOut* events are sent after calling *setTimeout(delay)*: the *TimeOut* event is sent *delay* milliseconds after the call. The timeout can be cancelled with *cancelTimeout()*. There can be only one timeout active at the same time. Therefore calling *setTimeout* a second time before the *TimeOut* event has been sent resets the timer to the new delay.

These events can be combined with an object type in order to specify that the event must have occurred when the cursor was on an object of this type. For example, the transitions "*Down on Document*" only fires when clicking on a file icon, and "*Enter on Folder*" only when the cursor enters a folder icon. The object under the cursor can be retrieved with the actions *GetDocument()*, *GetFolder()* and *GetWindow()*.

As an example, here is a state machine that displays a tooltip when the cursor is over a document icon for 300ms:



1. Draw the state machine that implements the standard drag-and-drop technique for moving a file icon onto the icon of a folder. The action *Move(file, folder)* moves the file into a folder.
2. Draw the state machine that implements the standard drag-and-drop technique for moving a file icon into an open window. Using the action *WindowOnTop(win)*, make it so that the window under the cursor goes on top when the cursor is in a window and does not move for 300ms. The action *Move(file, window)* moves the file to the window's folder.

We now consider the "spring-loaded folders" interaction technique: when dragging an icon, if the cursor is over a folder icon and does not move for 300ms, this folder opens into a window. The user can then place the file in that window, or move the cursor over a folder icon in that window and open it, etc. The action *OpenWindow(folder)* opens a window showing the content of *folder*.

3. Create a storyboard that illustrates this technique.
4. Draw the state machine that implements this technique.
5. Compare the "spring-loaded folders" technique with the standard technique of opening the target window first and then moving the file icon into it.

Answer part C questions on both sides of this page.

C. Course questions (7 points)

1. Explain the difference between an **algorithmic** and an **interactive** system:

2. Add a checkmark to the statements that are true:

- SketchPad supported videoconferencing
- The Xerox Star had windows and icons
- The first Macintosh was networked
- The first World-Wide Web browser supported editing pages

3. What is **change blindness**?

Why is it relevant to HCI? Give an example

4. Indicate the **type of task** (from the list of six basic interaction tasks seen in class) corresponding to each of these interactions:

Select the Copy command in the Edit menu: _____

Drag a file icon to the trash: _____

Click and drag to specify the position and size of a rectangle: _____

Click and drag to move a corner of a rectangle: _____

Use the scrollwheel when reading a document: _____

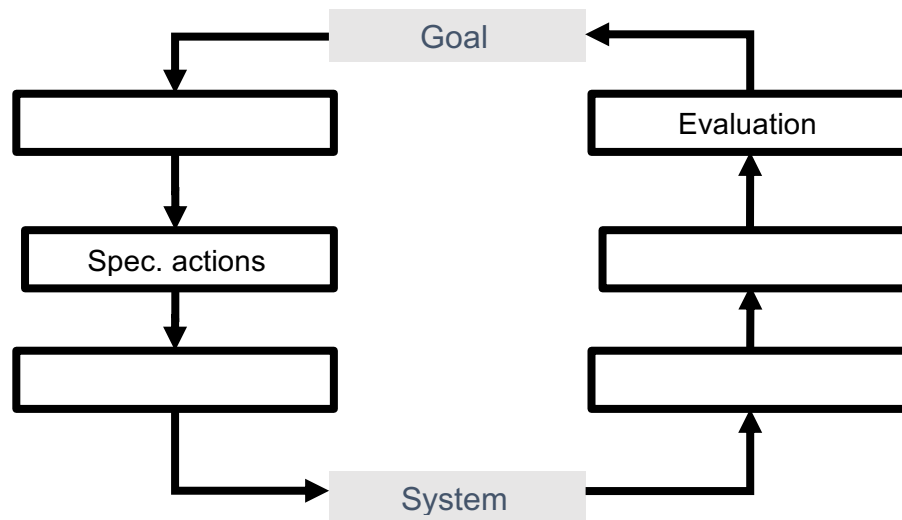
Set the number of copies in a Print dialog box: _____

5. Cite two techniques that **improve pointing** and explain why:

1. _____

2. _____

6. Add the missing stages in Norman's **seven stages of action**:



7. For each interaction, check one box according to whether it is based on **recognition** or **recall**:

<i>Recognition</i>	<i>Technique</i>	<i>Recall</i>
<input type="checkbox"/>	Menu	<input type="checkbox"/>
<input type="checkbox"/>	Gesture	<input type="checkbox"/>
<input type="checkbox"/>	Command line	<input type="checkbox"/>
<input type="checkbox"/>	Drag and drop	<input type="checkbox"/>