Exam question and sample answers for MSc HCI
Alan Dix

Sample Answer 1

[[Applied bookwork - need to think about appropriate visual cues]]

(i)   N.B. I will look for a careful distinction to be made between these answers and those of part (ii)
- NO semantic cues
- Stereo vision – better for miniature vs. bigger distinctions, but not so good for distinguishing normal from giant because of reduced angular distinction with distance.
- Blue hazing with distance (especially good for normal vs. giant)
- Occlusion effects – e.g. large person behind village would mean either person is giant and village is normal or person is normal and village is giant
- Some possibility of perspective effects *if* the landscape includes strong linear features

(ii)   Good students will notice that assumptions have to be made about the users' size and rate of movement in the virtual world.
- Various parallax effects due to sideways movement to give comparative distance between different villages
- Rate of approach as one moves towards villages
- Height of head – looking *down* on miniature village
- Comparative size of user's own virtual hand and body (where present) as the villages are approached

(iii)   The critical issues will have already been discussed above – this is reapplying them in a different context.

- Difficult because shapes are abstract and different sized spheres, mean size cannot be used as an indication of distance.
- Interactive 3D display allows effects like movement parallax etc., as in part (ii)

Sample Answer 2

(i)   [[mainly bookwork]]
         Not interested in the language used here, just the concepts
- Application Interface Model – this accesses the API of the underlying functions of the system
- Dialogue – manages the ordering of things, what the user is allowed to do when, how the system responds, but not the actual detailed content, works on user commands at an abstract level (e.g. 'save')
- Presentation – determines how things actually appear on screen, graphical layout etc., and the translation of raw user input (e.g. click at 372,413) into abstract actions (e.g. user clicked the 'save' button)

(ii)   [[mainly bookwork]]
         The switch
- The switch is there to allow rapid semantic feedback
- Controlled by the dialogue component, but not mediated by it.
- This is because things like line drawing require direct connection between low level user actions and the underlying functional core.
- May use example of dragging file over trash can/folder (with highlight) as opposed to other file (no highlight)

(iii)   [[design]]
         Some students may talk in terms of particular technologies (CGI etc.) other in more general terms (server-side processing), I'm interested in the overall architecture client/server trade-off, so either approach is acceptable.
- They ought to take into account the slow rate of update here. For fixed view periodically generating pages offline from a database would be sensible approach and then using standard web server.
- The hotel details could managed by web transaction as each hotel is clicked, or could use client side (Java or Javascript) – I'm looking for mention of different rate of feedback for the user.
- In stage 2 more dynamic database access is required, not because of the rate of update which is still the same, but because of the more complex queries.
- Students may use pictures like below and the better students may draw parallels with the Seeheim model.
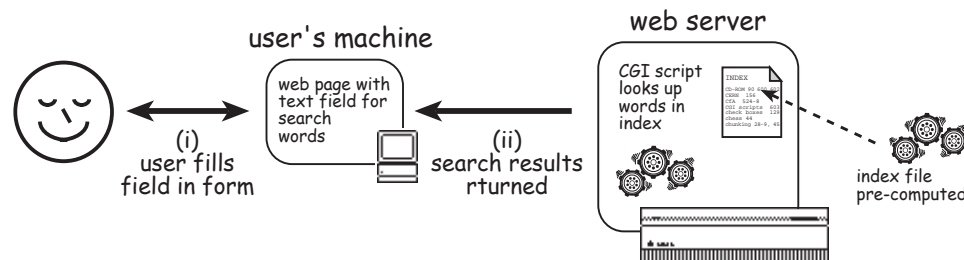
Sample Answer 3

[[analysis/design]]

(i)   This part is mainly a fleshing out of the 'find number subtask. Students may choose to use the textual form of HTA and this is acceptable.
  - Simple answer for plan 1 would be one or other, but some students should think a bit more about what this is really like for users.

```
                                      ┌─────────────┐           ┌─────────────┐
                                      │ 1.          │           │ 2.          │
                                      │  find number│           │  make call  │
                                      └─────────────┘           └─────────────┘
                                        plan 1.                      ┊
                                          try 1.1                    ┊
                                          if number not found do 1.2 ┊

                    ┌──────────────┐                ┌──────────────┐
                    │ 1.1  use     │                │ 1.2   use    │
                    │ address book │                │  directory   │
                    │              │                │  enquiries   │
                    └──────────────┘                └──────────────┘
                                                       plan 1.2
        possibly describe                                2.1 , when connected 2.2
        sub-tasks here too.                              when finished call 2.3

              ┌──────────┐     ┌──────────────┐    ┌──────────┐   ┌──────────┐
              │ 1.2.1    │     │ 1.2.2        │    │ 1.2.3    │   │ 1.2.4    │
              │ dial 192 │     │ ask for number│   │   talk   │   │ end call │
              └──────────┘     └──────────────┘    └──────────┘   └──────────┘
```
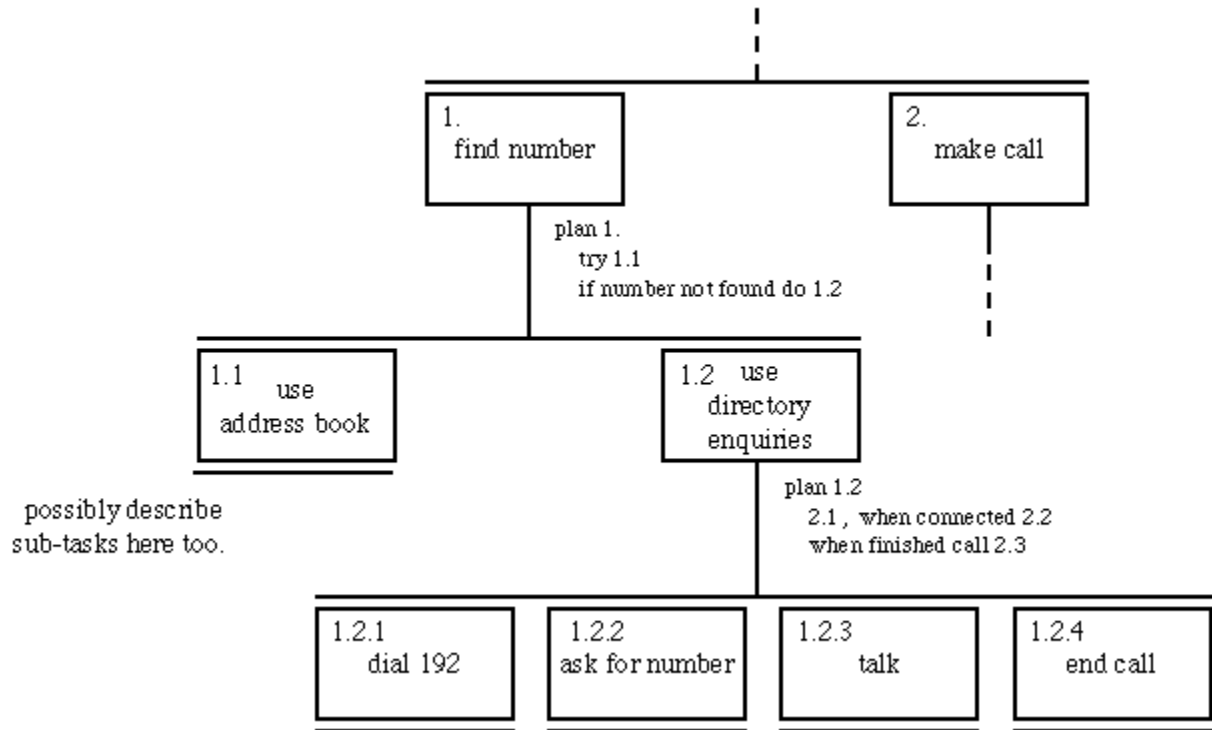
(ii)   Main subgoals below,..
- but students should either flesh out 2.1 and 2.2 or justify that this is unnecessary

```
                        ┌─────────────┐
                        │   phone     │
                        │ 0. a friend │
                        └─────────────┘
                              │
                         plan 0.
                           1. then 2.
              ┌───────────────────────────────┐
        ┌───────────┐                   ┌───────────┐
        │ 1. lookup │                   │ 2.  have  │
        │ shortcut  │                   │conversation│
        └───────────┘                   └───────────┘
              │                               │
          plan 1.                          plan 2.
        1.1, 1.2, 1.3 in order            when connected 2.1
                                          when finished call 2.2
    ┌─────────┬─────────┬─────────┐     ┌─────────┬─────────┐
 ┌────────┐┌────────┐┌────────┐      ┌───────┐┌────────┐
 │2.1 get │││2.2     │││2.3 enter│     │2.1    │││2.2     │
 │to      │││select  │││shortcut │     │talk   │││end call│
 │'phone  │││recall  │││pos      │     │       │││        │
 │book'   ││        ││        │      └───────┘└────────┘
 │menu    ││        ││        │
 └────────┘└────────┘└────────┘
```
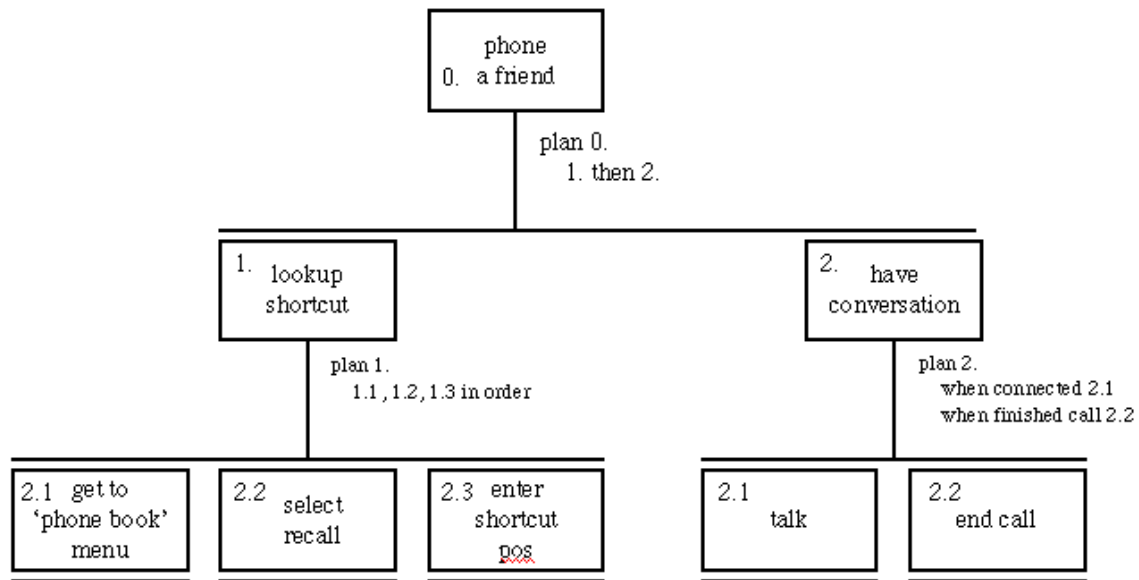
(iii)   Big difference is that the dialling of the number gets folded into finding it.
- One solution is to transform the view of the original tasks (see below)
- Alternatives may be to have a conditional sub taks 'dial number if necessary'
- May also note here or earlier that waiting may be worth including as sub-task
- But should be asking whether these are sensible tasks from a user perspective
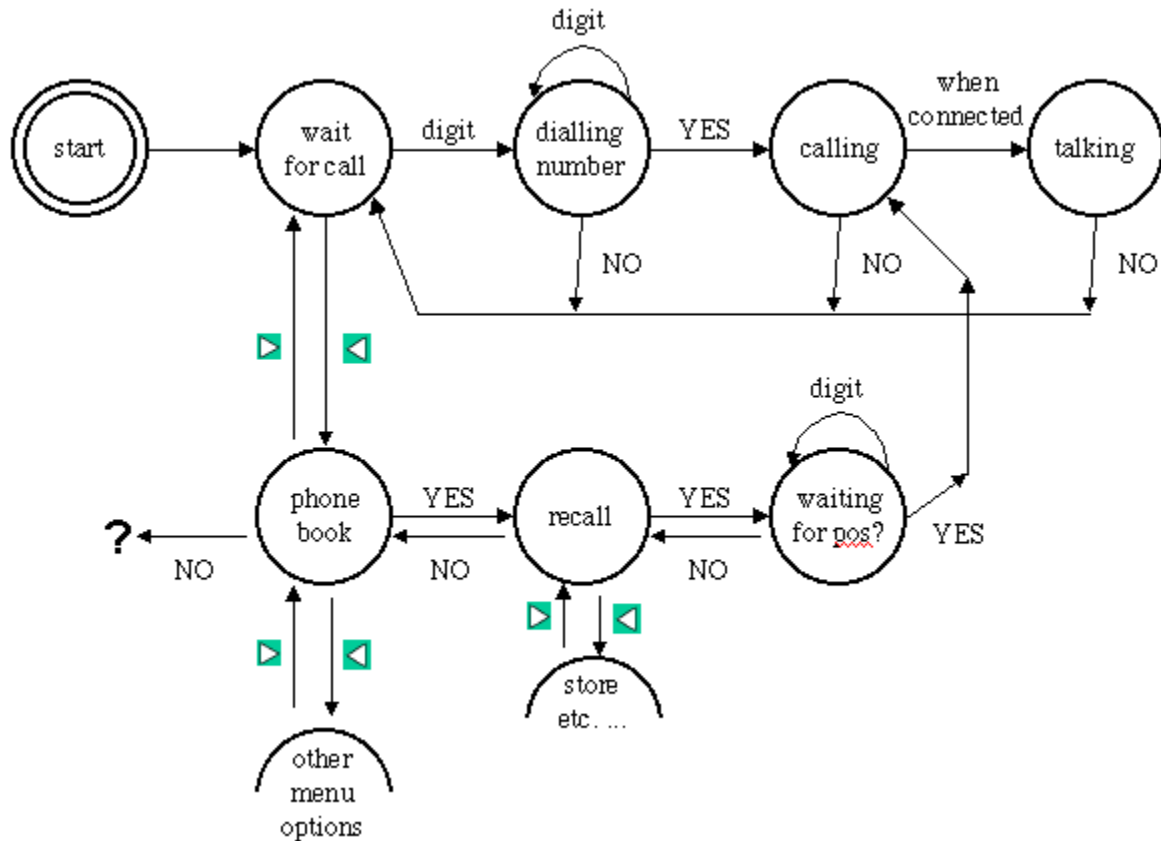
```
                        ┌─────────────┐
                        │   phone     │
                        │ 0. a friend │
                        └─────────────┘
                              │
                          plan 0.
                            1. then 2
                            when connected 3.
        ┌──────────────┬──────────────┬──────────────┐
  ┌───────────┐  ┌───────────┐  ┌───────────┐
  │ 1.  find  │  │ 2. wait   │  │ 3.        │
  │ and dial  │  │ for       │  │ make call │
  │ number    │  │ connection│  │           │
  └───────────┘  └───────────┘  └───────────┘
        │                              ┆
     plan 1.                           ┆
       1.1 or 1.2 if you have the number
       if not, or either fails 1.3
    ┌──────────┬──────────┬──────────┐
 ┌────────┐┌────────┐┌────────┐
 │1.1 use │││1.2 use │││1.3 use │
 │address │││mobile's│││directory│
 │book    │││phone   │││enquiries│
 │and dial│││book    │││and dial│
 └────────┘└────────┘└────────┘
```
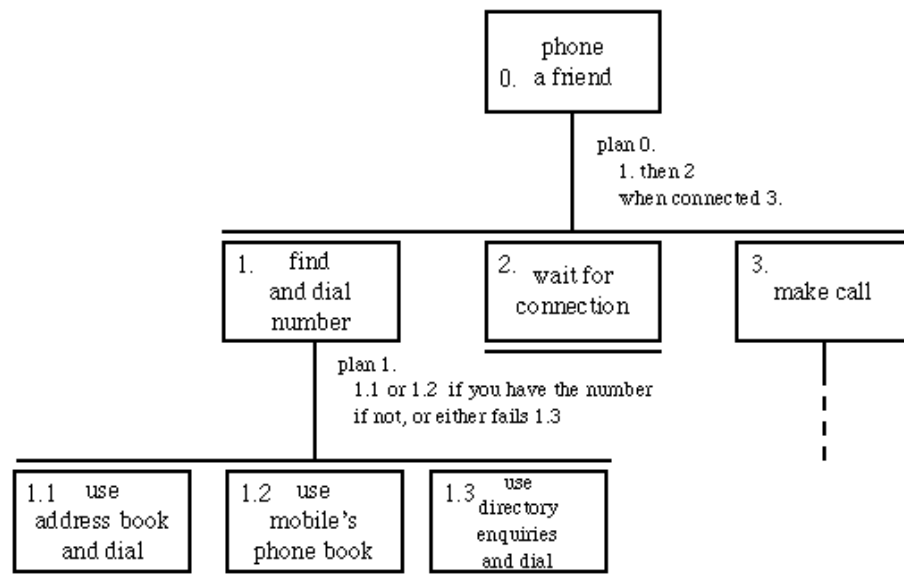
Sample Answer 4

** N.B. change marking scheme to 4, 8, 8 **

[[analysis/design]]

(i)   As well as formal analysis, this question asks the students to map back from
        formal STN to the meaning of actions in given situations for the user. Note that
        I'm not looking for oarticular suggestions for 'corrections' to the STN. More how
        they think about it.
        • Missing 'YES' and 'NO' transition missing from 'wait for call' state - probably
          ignore
        • No digit transitions in states 'calling' and 'talking' - any experience with online
          phione systems suggests that it is necessary to transit digits to far end when in
          'talking' state, for 'calling' state could ignore, (or buffer and then send to far
          end allowing 'typeahead' for phone-based systems. In terms of the STN all
          would be no state change - the difference is in the semantics.

(ii)   The scenario has lots of gaps. The STN shown here is built from the info in the
        scenario, 'world' assumptions (e.g. left and right arrows are inverses of one
        another) and 'sensible' adding of where extra menu/submenu items would be
        found. Again answers will differ. Critical things for marking:
        • Is it consistent with scenarios?
        • Are additions justified and sensible from a user perspective?

digit

when connected

start → wait for call → digit → dialling number → YES → calling → talking

wait for call → dialling number: digit (self-loop on dialling number)
dialling number → NO
calling → NO
talking → NO

phone book → YES → recall → YES → waiting for pos? → YES

? ← NO — phone book — YES → recall — NO
recall ← NO — waiting for pos?

digit (self-loop on waiting for pos?)

YES

store etc. ...

other menu options

(iii)   Answer clearly needs to be consistent with answer to (ii) and with scenario:

- State description (here in C syntax), may have one 'state' variable or one for menu and one for sub-menu one for state in sub-menu. Here just have single (most likely solution)

```
enum { WAIT, DAILING, CALLING,
       TALKING, PHONE_BOOK,
       PB_RECALL, PB_RECALL_POS,
       PB_STORE, .… } mainstate;  // which bubble
int current_number;  // or stored as string
int[] phonebook;  // as this is fixed in recall scenario
                  // may not be considered by many students
```

- We used a tabular form for this in lectures, but a more discursive style as below would also be good.

```
mainstate= WAIT,  current_number=any,
phonebook={0121456, 043267, 084534, … etc… // doesn't change
```

       C.1   Andy presses the left arrow key ◄

```
mainstate= PHONE_BOOK,  current_number=any
```

       C.2   The phone display says 'phone book'.
       C.3   He presses 'YES'

```
mainstate= PB_RECALL,  current_number=any
```

       C.4   The phone display says 'recall'

C.5   He presses 'YES'

`mainstate= PB_RECALL_POS,   current_number=0`

C.6   The phone display says 'pos?' (requesting position in phone book)

C.7   Andy enters the shortcut digit (3) for his friend.

`mainstate= PB_RECALL_POS,   current_number=3`

C.8   The digit appears on the phone display

C.9   Andy presses 'YES'

`mainstate= CALLING,   current_number=084534`

C.10   The phone display says 'calling', as in A.4 on