Master – 28/4/03

I&C 3.1

1.
(a) 10 marks, application.
(b) 10 marks, application.

Obviously there are many variations for valid answers to this question, not so much with the identified design issues, but most certainly with the design. One possible answer is given below.

(a)
Design Issues;
Typical user: infrequent user of target system. Novice computer user. Unfamiliar with technical vocabulary of gardening terminology.

Typical Task: initial requirements are only vaguely consciously formed, if at all. User needs to become familiar with what is available. May have some idea of colour, size, soil type and blooming season. Will gradually formulate requirements. Need for a basket to place potential purchases.

Presentation: Since user's requirements are often unknown, some form of opportunistic browsing may be appropriate, to simulate the viewing that takes place as people wander around a garden centre. Possibly use a slow moving sequence of plant images, the content of the 'stream' being influenced by selections of cost, size, colour, etc. A zoom facility should be used with care, and should always retain a view of context. There should be no tree structure (sic).

Representation: Usually the actual image of the plant will be needed. Symbolic encoding may be a possibility, but should be used with extreme care.

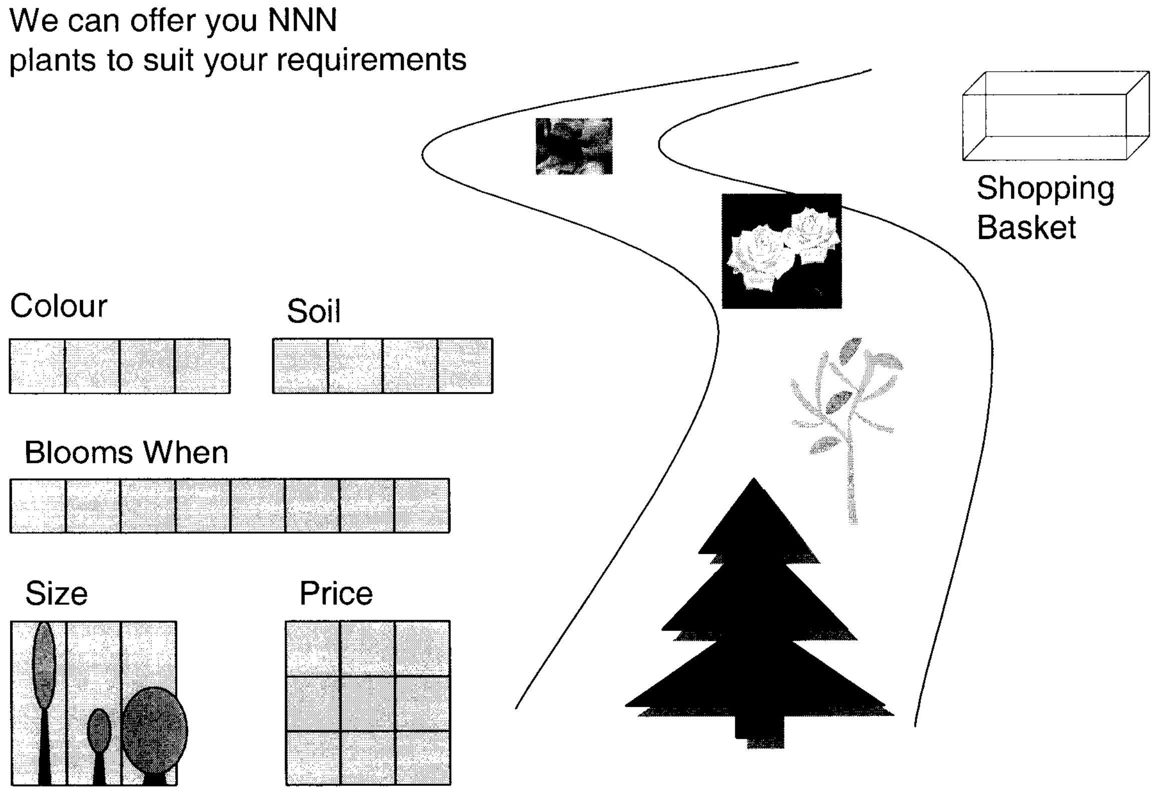
Interaction: By touch screen (public access, chance of vandalism, many people cannot operate mouse or joystick). Allow simple specification of size (3 levels), colour of bloom (5 choices), blooming month (up to 12) and perhaps soil type and amount of sunshine/shade. Use animation to show affordances (moving hand, selection of attributes, placement of plant in basket)

(b)
First trial design: see overleaf, plus comments.

Areas that need more consideration:
Needs some way in which the user can receive recommendation about changes to their blooming period requirement. Perhaps parallel indication of best match.

Requires careful choice of background colours to help user to understand affordances.

Expert users agents to supply detailed verbal guidance.



We can offer you NNN
plants to suit your requirements

Key points:
Top left: provide some indication of relevant database

Bottom left selectable colours, easily interpreted, size selector should have humans reference size, cost selector allows one selection (default set by FlowerPower)

Middle, slow moving stream of items, entering at top, falling off at bottom, getting bigger
As it progresses, showing plants/trees etc that satisfy requirements. Touch on image to stop stream, pop-up additional information

Top right: touch, drag and drop to add to shopping basket

2.
(a) 4 marks, Bookwork if background reading done
(b) 4 marks, Bookwork if background reading done
(c) 8 marks, Application (in particular coursework)
(d) 4 marks, Bookwork

There are some easy marks on offer here PROVIDED that the appropriate effort has been invested in additional background reading and actually doing the coursework, with a free gift at the end.

(a)

- **what are guidelines?**
  - expressions of principle
    - 'know thy user', 'design for error', ...
  - design rules
    - 'provide an undo command', 'always issue warning message', ...
  - rules of thumb
    - e.g. minimum font sizes, font styles, ...
- **why have guidelines?**
  - user interface design is expensive
    - help establish requirements, make design decision, in evaluation
  - guidelines promote conformity
    - consistent 'look and feel', familiar corporate image
  - guidelines offer protection
    - satisfy corporate directives, reject only with good reason

(b)

- **What are standards?**
  - regulated requirements (standards by law)
    - legal requirements: EC imposed quality requirements, safety
    - contractual requirements: military contracts
  - prescribed norms (standards by agreement of International body)
    - ISO standards
    - ASCII character set
    - programming language syntax and semantics
  - de facto norms (standards by mutual agreement)
    - QWERTY keyboard, car steering wheel, KQML, (plug wiring?)
- **Why have standards?**
  - demand for interoperability, quality and safety
  - when conformity needs to be measured

(c)
The insight required here is to specify the ten heuristics as guidelines rather than heuristics, and indicate with example how it supports ease of use, ease of learning, responsiveness, permissiveness and consistency.

Visibility of system status, supports responsiveness
Keep users informed of what is going on
E.g. status checks, progress bars, animated icons, throbbers and so on.

Match between system and real world, supports consistency, learning

Speak user's language, use real world phrases, familiar to user, follow real world conventions
E.g. mail system refers to mailboxes, messages, uses envelope icons, etc

User control and freedom, supports permissiveness, ease of use
Easy to correct accidental operation, emergency exits, support undo and redo
E.g. undo/redo menu commands, installer systems

Consistency and standards, supports consistency, ease of learning
Follow platform conventions, same word does same thing
E.g. different dialogues implemented by different programmers on OK, Cancel, Apply

Error prevention, supports ease of use
Stop problems occurring is better than good error messages
E.g. putting arbitrary iconic window controls next to each other on window pane

Recognition rather than recall, supports ease of use, learning, permissiveness
Make objects and actions visible, don't force users to remember information from one step of a dialogue to another, instructions for use
E.g. windows XP menus

Flexibility and efficiency of use, supports permissiveness, ease of learning
Cater to experienced and inexperienced users, allow user to customise
E.g. Keyboard Accelerators

Aesthetic and Minimalist design, supports ease of responsiveness, consistency
Dialogues should not contain more information than needed
E.g. animated characters that have no functionality except to steal cpu cycles from user task

Recognise, diagnose and recover form errors, supports responsiveness
Errors explained in plain language, not error codes, indicate problem and suggest solution
E.g. compare compilers vs. 404, Exception at 000000000FB

Help and Documentation
Easy to search, task-oriented, list steps
E.g. on-line help

(d)
Controls vs controls: temporal, functional and frequency
Controls vs displays, spatial, movement and cognitive compatibility

3.
(a) 10 marks, Bookwork plus
(b) 6 marks, Bookwork plus
(c) 4 marks, Explanation

While the bulk of this question is based on bookwork, the requirement for this question is that the discussion must contain a bit more than mere repetition of the lecture notes, as contained here, but also to explain it with supporting material as presented in the lectures themselves. The additional constraint is the wide scope of this question means moving quite fast to offer a full answer.

(a)

Objective

➢**objective: to assess performance objectively and identify causes of problems**
➢**method: assessment is based upon watching the use of a system**
➞simulation, prototype, or final product
➞"simulated" users or representative actual users
➞simulated tasks or actual tasks
➞simulated context or actual context
➞

Method

➢**a systematic approach to the evaluation is required:**
①**decide scope of the study**
➞to get enough of the right sort of data
②**plan & implement observations**
➞using appropriate data collection techniques
③**classify & analyse behaviour**
➞according to the objectives of the evaluation
④**evaluate performance**
➞according to the requirements of the evaluation

Step 1: Scope

➢**clear specification of the purpose of the study; i.e. stating the requirements**
➞e.g. informal identification of user problems
➞formal comparison of two input devices
➢**issues of scope include**
➞what criteria are going to be used in the evaluation
–performance, errors, usability
➞what sort of report is required
–summative or formative
➞what sort of user should be considered
➞what sort of tasks should be considered
➞what data should the evaluation be based on
➞how detailed the conclusions need to be
➞degree of control

Step 2: Planning & Implementing Observations

➢**data collection methods include:**
➞evaluator's direct observations
➞system monitoring (e.g. keystroke logs)
➞audio/video recording

➢**techniques for exposing mental processes**
➟verbal protocols
–concurrent
–retrospective
➢**issues in data collection**
➟interference with behaviour of interest
➟availability of time and resources

Step 3: Classification & Analysis of Behaviour
➢**seek to classify pattern(s) of behaviour which can be interpreted as meaningful actions in completing a task**
➟e.g. clicking with mouse, entering text, looking at manual/help
➢**classification may be supported by video analysis**
➟task based analysis
–determine how user tackled the given tasks to highlight major difficulties
➟performance based analysis
–performance measures from data collected
–e.g. frequency/time of correct task completion, use of commands, frequency of errors, time for cognitive actions (thinking, reading, etc.)
➢**issues in behavioural classification and analysis**
➟granularity of analysis
➟can be laborious and time consuming
➟statistical significance

Step 4: Evaluate Performance
➢**report on performance**
➟e.g. times, number of tasks successfully completed, number of errors
➢**comparison**
➟between desired and actual performance
➢**for formative evaluation (only)**
➟suggested explanation for observed performance inadequacies attributed to user problems with the interface
➟data supporting recommendations for re-design
➢**contents of an evaluation report**
➟'mission', scope, assumptions
➟method (plan), requirements & resources
➟description of observations (i.e. the data obtained)
➟classification and analysis
➟conclusions

(b)
At the conclusion of an observational evaluation session, the data subject may be interviewed or given a questionnaire to fill in. This offers the additional opportunity to get a subjective assessment of the system, in conjunction with the observed behaviour.

➢**method: users are requested to provide information about their usage of a system**
➟interviews (data collected tends to be qualitative)
–direct communication between user and evaluator
–range between structured and unstructured

▪questionnaires (data collected tends to be quantitative)
–standardised format of data relatively easy to access and analyse
–potential for large scale data collection and statistically significant results
➢**may have a number of functions during development:**
▪as input to requirements (assessing performance of existing system)
▪as input to design (assessing performance of prototype)
▪as input to redesign (assessing performance of new system)

Pros and Cons: method 1
➢**Interviews**
▪advantages
–sensitive to evaluator's requirements
–fosters user-developer relationship
–does not require system to be present
▪disadvantages
–users can only report on direct experience
–labour-intensive for both users and evaluator
–relies on communication skills of evaluator
–evaluator may introduce bias
–**Questionnaires**
–low cost means of acquiring lots of data in standardised form
–not necessary for evaluator to be present
–standard questionnaires are available which can be tailored
▪disadvantages
–users can only report on direct experience
–response rate is often low or biased
–skill required to develop precise unambiguous questions
–evaluator may introduce bias

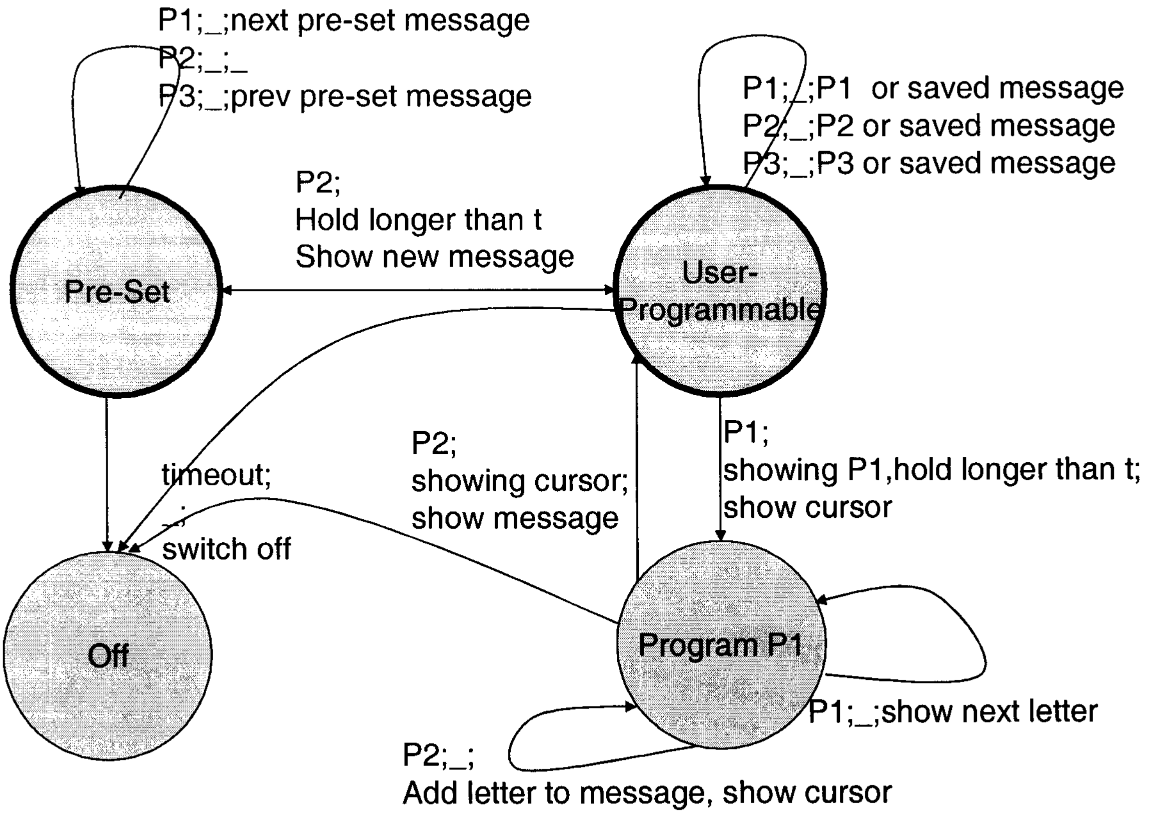
(c)
observational: real users, real system
user: real user abstract system
analytic: abstract user, abstract system
specialist: real system abstract user

Dependence on current place in development life cycle, evaluation criteria (e.g. formative vs summative)

(4)
(a) 8 marks, Application
(b) 4 marks, Application
(c) 8 marks, Application

Need to understand and apply material to do question. Part (a) not especially onerous, except for having to untangle the given description (and trust me this is much clearer than that which actually comes with a skyliner). Part (b) and (c) require some insight into the process, a little bit of awareness for the input mechanism, i.e. a jog dial of some sort. Was discussed in the lectures.

(a)



Notation: circle is state, arc is action action;condition; effect
Note two start states
Note two missing states for Program P2 and Program P3, but identical to Pro gram P1

(b)
Various problems, including:
Actions need timers
No action causing a transition
Same action from every state
Reference to counters, timers etc., there isn't much of an interface to speak of
Note 'combinatorial explosion of states' is not a difficulty in this context

(c)
Redesign involves replacing the three buttons with a rotating wheel which can be pressed, i.e. a jog dial.

In pre-set mode:
Rotate one way shows next message, other way shows previous message
Press just once, toggle mode.

In user-programmable mode
Rotate one way shows next saved message, other way shows previous saved message
Press and hold, until LEDs go out and come on again, erase and start entering message:

Rotate one way shows next letter, other way shows previous letter
Press just once to save letter
Press twice to save message.

Note press to toggle mode, press and hold to program slot is considered safer than the other way round in terms of preventing accidental operation.

5.
(a) 8 marks, Application
(b) 4 marks, Application
(c) 8 marks, Application

This question is similar in rationale to Q4.

(a)
For the user, the basic problem is that at each step, they are forced to make a choice, the scope and consequences of which are not clear.

This leads into standard problems of issues in menu design, namely

- **Category Label Ambiguity**
  - **It is not clear whether some items should be in one category or another, e.g. movies cross categories**
- **Structure: Breadth vs. Depth**
  - **The lack of screen space and hierarchical thinking (reflecting the system data structures) has forced the designer into a deep structure which is inappropriate for the application**
- **Errors vs. Exploration**
  - **Allowing selection of a movie which then cannot be viewed (also putting personal prejudice into design)**
- **Navigation**
  - **Issue of context: in so many deep screens the user can become lost**
- **Hierarchy**
  - **The hierarchy is artificial, and there is actually no natural sequence of selection**

For the passenger in the seat in front, having the guy behind repeatedly tapping the back of your seat would be inducement to air rage...

(b)
The superficial appeal is that JSD are good for representing turn-taking dialogues. The drawback here is that the apparently sequential nature of the interaction is actually sucking the user deeper and deeper into the hierarchy, with corresponding loss of context, no provision for going back, and so on.

(c)
The idea is to present all possible choices in a single continuous 'carousel' with minimum selection type interaction.

Action: rotate wheel
Condition: none
Effect: Show next item

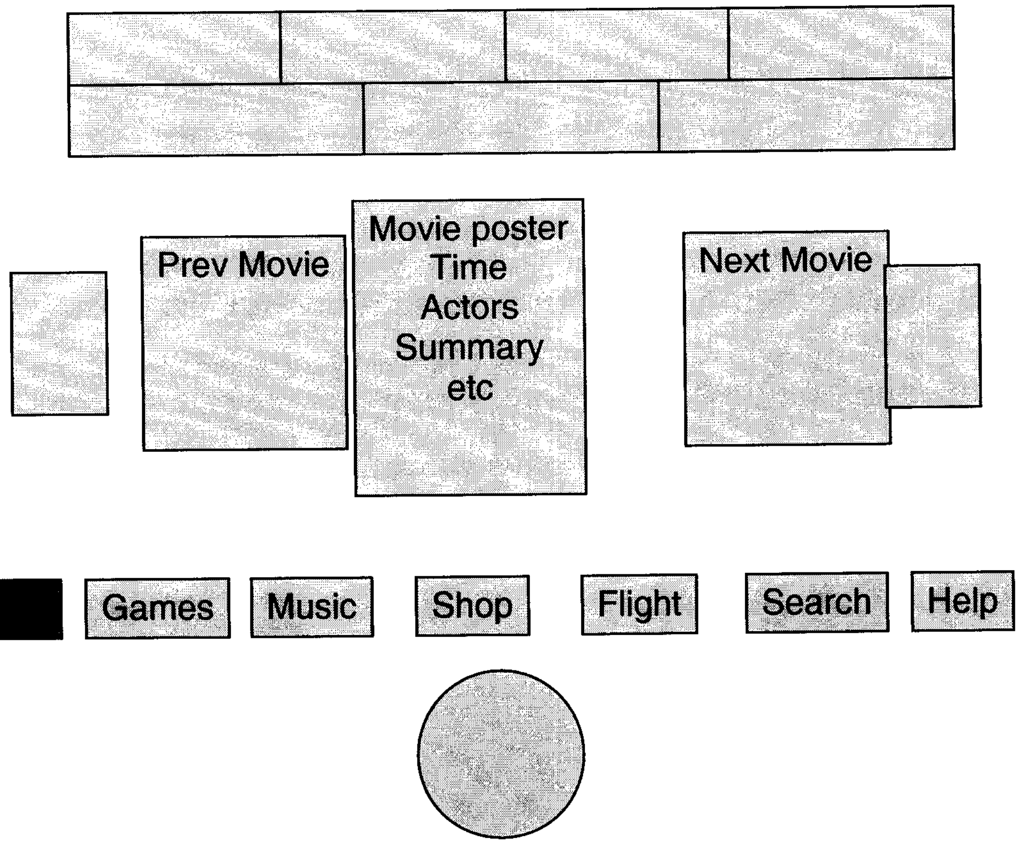
Action: Press touchscreen
Condition none
Effect: Select Item

Action: Press Menu item
Condition: none

Effect: jump to corresponding part of carousel

Copious with information, parsimonious with choice
Plenty of information about choice under picture, including categories at top of screen
Menu of main choices along bottom
Only makes one choice when ready

Cater for experienced users by accelerators in form of menu choices
Search facility allow customise carousel, for example.



Note from bottom:
Single physical rotating device, advances options left or right according to counter or clockwise rotation (movement compatibility)
Menu of main choices for fast selection, highlight changes according to front item
Selectable menu items on carousel, using RSVP
Category information , selectable in search mode
Items that are unavailable will not be displayed

6.
(a) 4 marks, Bookwork
(b) 6 marks, Bookwork + Background reading
(c) 10 marks Application

(a)
These are interfaces that may be found at the extreme end of the spectrum of applications in the domain of HCI, (Human Computer Interaction). They are interfaces, that by their nature, their applications and their modes of operation, are often innovative and tend to operate at the edge of conventional interface technology. Examples include expressive ballet shoe, haptic suit, sensed data glove, etc, as below

(b)
There are many possible answers, but here are 6 discussed in the lectures.

Device 1. Pressure Sensing Pads:
Piezoelectric pressure sensing pads are combined with other sensing devices to track the dynamics of human movement (e.g in an "Expressive Shoe").

Application 1. Expressive Ballet Shoe:
The Expressive Ballet Shoe is one such device. The data from pressure sensing pads (combined with other sensing devices) embedded in the shoe are reported continuously by wireless transmission in real time to a synthetic orchestra which acts as a continuous accompaniment to the performance of the ballet dancer. See "Expressive Footwear" reference in the notes.

Device 2. Vibro-tactile transducers:
Arrays of transducers woven into in a full-body suit transmitting continuous vibro-tactile data from an attached computer system to the human subject wearing the suit.

Application 2. The Haptic Suit:
A computer wearable interface, which exploits the delicate sensory channel of human touch to augment the audio appreciation of music. Embedded transducers in the suit act as vibro-tactile sensors complementing the musical experience conveyed through the head phones. (see ref: to the Haptic Suit in the "Extreme Interface" article in the attached notes)

Device 3. Bio-feedback Sensor:
Sensors attached to an operator's skin and having the capacity to report ongoing variable changes in the emotional state of the operator, e.g in interactive Affective Mind Games.

Application 3. Affective Mind Games:
In Mind Games, the speed of the racing dragons is controlled via bio-feedback sensors attached to the players' fingers. The player achieving the highest level of relaxation communicates this to the game system and thereby elicits a winning performance from his/her dragon. See "Extreme Interface" article (featuring Mind Games) in the notes.

Device 4. Tracked LED:
Arrays of light emitting diodes being continuously sensed in the presence of photo diodes is an effective movement tracking technique in HCI.

Application 4. The Sensed Data Glove:
One application of this tracking technique is the sensed data glove, in which the movement and gestures of the hand act as input to a real time computer animation system. In the case study reported in the attached notes "Gestural Input", the Data Glove idea has been expanded to become a full-body Data Suit acting as a real time animation input medium.

Device 5. Touch-sensitive Screens:
Touch-sensitive screens reporting operator's finger contact and movement as input to multimedia information systems.

Application 5. The Electronic Coffee Table:
One innovative application for the screen (above) is the Electronic Coffee Table case study in which such a screen acts as an innovative interface to an agent-based multimedia system for a local neighbourhood community. See paper "The Electronic Coffee Table" in the attached notes.

Device 6. Precision Strain Gauges:
As HCI devices, sensitive force-sensing gauges can report variable forces being applied to surfaces by an operator.

Application 6. Hyperviolin:
Hyperviolins are members of the stringed family Hyperinstruments in which precision force-sensing strain gauges and other monitoring devices report the actions of a virtuoso to a linked synthetic orchestra, thereby extending and augmenting the performance of the soloist. See refs. to "Hyperstrings" & "Extreme Interfaces" in the attached notes.

(c)
Huge range of possible answers, but choose from sensible explanations of: embodied agents, affectiveware, monitoring preferences, advertising prompts, electronic store guides, sensing movement patterns, sentient precinct, MAPPA-like kiosks, LIME-like coffee tables, loyalty cards, alerting for presence of friends, to name 11...