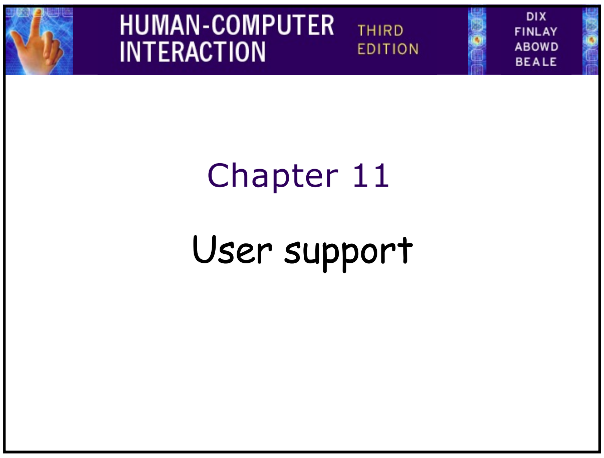
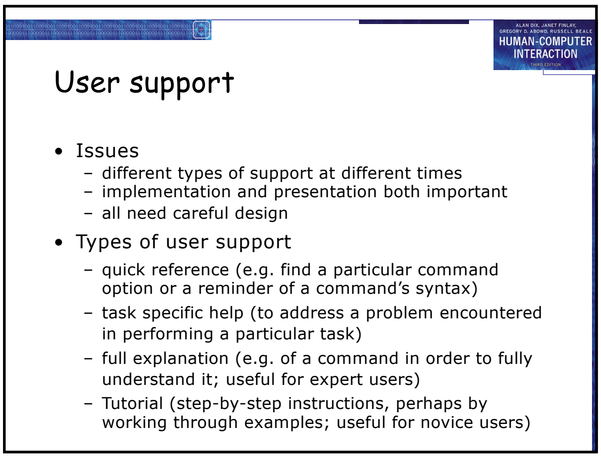


Chapter 11

User support

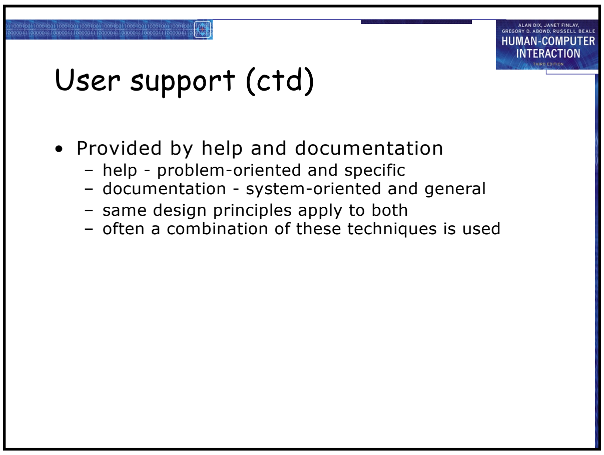
1



User support

- Issues
  - different types of support at different times
  - implementation and presentation both important
  - all need careful design
- Types of user support
  - quick reference (e.g. find a particular command option or a reminder of a command's syntax)
  - task specific help (to address a problem encountered in performing a particular task)
  - full explanation (e.g. of a command in order to fully understand it; useful for expert users)
  - Tutorial (step-by-step instructions, perhaps by working through examples; useful for novice users)

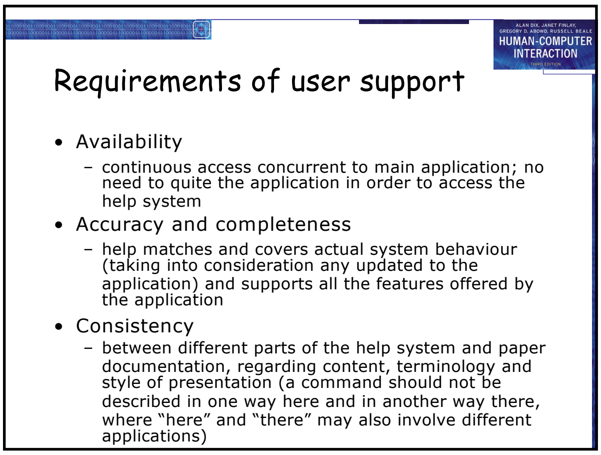
2



User support (ctd)

- Provided by help and documentation
  - help - problem-oriented and specific
  - documentation - system-oriented and general
  - same design principles apply to both
  - often a combination of these techniques is used

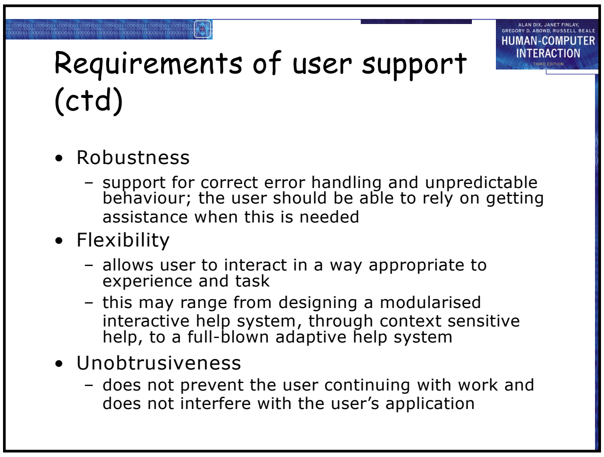
3



Requirements of user support

- Availability
  - continuous access concurrent to main application; no need to quite the application in order to access the help system
- Accuracy and completeness
  - help matches and covers actual system behaviour (taking into consideration any updated to the application) and supports all the features offered by the application
- Consistency
  - between different parts of the help system and paper documentation, regarding content, terminology and style of presentation (a command should not be described in one way here and in another way there, where "here" and "there" may also involve different applications)

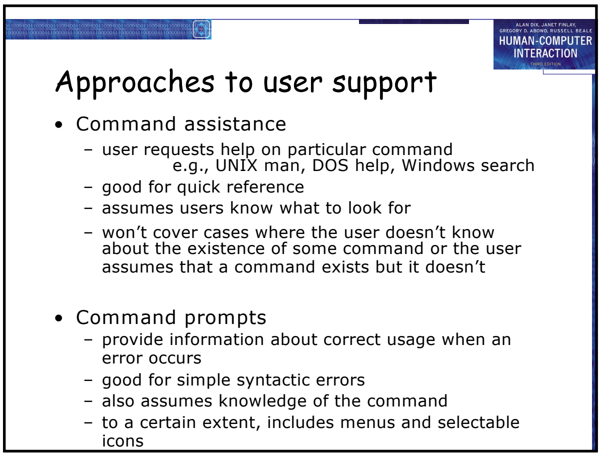
4



Requirements of user support (ctd)

- Robustness
  - support for correct error handling and unpredictable behaviour; the user should be able to rely on getting assistance when this is needed
- Flexibility
  - allows user to interact in a way appropriate to experience and task
  - this may range from designing a modularised interactive help system, through context sensitive help, to a full-blown adaptive help system
- Unobtrusiveness
  - does not prevent the user continuing with work and does not interfere with the user's application

5



Approaches to user support

- Command assistance
  - user requests help on particular command
    e.g., UNIX man, DOS help, Windows search
  - good for quick reference
  - assumes users know what to look for
  - won't cover cases where the user doesn't know about the existence of some command or the user assumes that a command exists but it doesn't
- Command prompts
  - provide information about correct usage when an error occurs
  - good for simple syntactic errors
  - also assumes knowledge of the command
  - to a certain extent, includes menus and selectable icons

6

Approaches to user support (ctd)

- Context sensitive help
  - help request interpreted according to context in which it occurs, e.g. tooltips and web page rollovers
- On-line tutorials
  - user works through basics of application in a test environment (in a self-paced mode)
  - can be useful but are often inflexible, as they have no knowledge of the particular user
- On-line documentation
  - paper documentation is made available on computer
  - continually available on-line (paper versions tend to get lost)
  - can be difficult to browse
  - hypertext used to support browsing

7

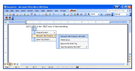
Approaches to user support (ctd)

- Simple guidelines for on-line documentation
  - use clear structure with headings to provide signposting
  - organize information according to user tasks
  - keep sentences short, to the point and jargon free - use simple but unpatronizing language
  - set out procedures in order and number steps - highlight important steps
  - use examples where possible
  - support searching via an index, contents, glossary and free search
  - include a list of error messages
  - include Frequently Asked Questions (FAQ) with clear answers

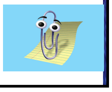
8

Wizards and Assistants

- Wizards
  - task specific tool leads the user through task, step by step, using user's answers to specific questions
  - example: resumé
  - useful for safe completion of complex or infrequent tasks
  - constrained task execution so limited flexibility
  - must allow user to go back and forth and provide a progress indicator



- Assistants
  - monitor user behaviour and offer contextual advice
  - can be irritating e.g. MS paperclip
  - must be under user control e.g. XP smart tags



9

Adaptive Help Systems

- Use knowledge of the context, individual user, task, domain and instruction to provide help adapted to user's needs, by means of monitoring the activity of the user and constructing a model for him
- Problems
  - knowledge requirements considerable
  - who has control of the interaction?
  - what should be adapted and what will be the result of the adaptivity?
  - what is the scope of the adaptation (possibly beyond the application level)?

10

Knowledge representation: User modeling

- All help systems have a model of the user
  - single, generic user, as understood by the designer of the system (non intelligent)
  - user-configured model, e.g. browser or email preferences (adaptable)
  - system-configured model (adaptive): requires setup time during which the user has a general default system (which will eventually "learn him") but the user is not required to build the model

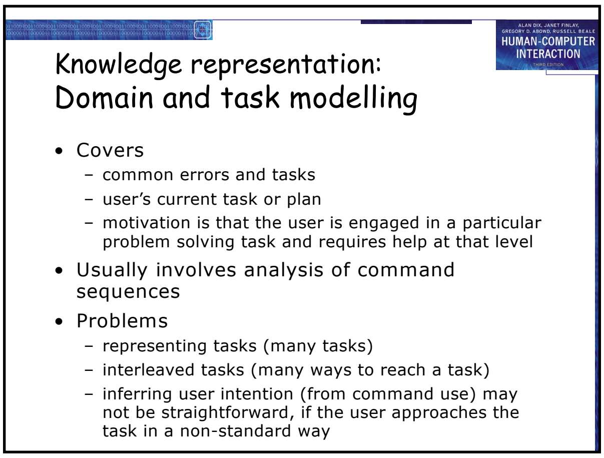
11

Approaches to user modelling

- Quantification
  - user moves between levels of expertise
  - based on quantitative measure of what he knows (e.g. to move from level 1 to level 2 the user has used effectively commands X & Y and help has not been accessed for 3 days)
- Stereotypes
  - user is classified into a particular category (e.g. novice and expert)
- Overlay
  - idealized model of expert use is constructed
  - actual use compared to ideal
  - model may contain the commonality or difference

Special case: user behaviour compared to known error catalogue

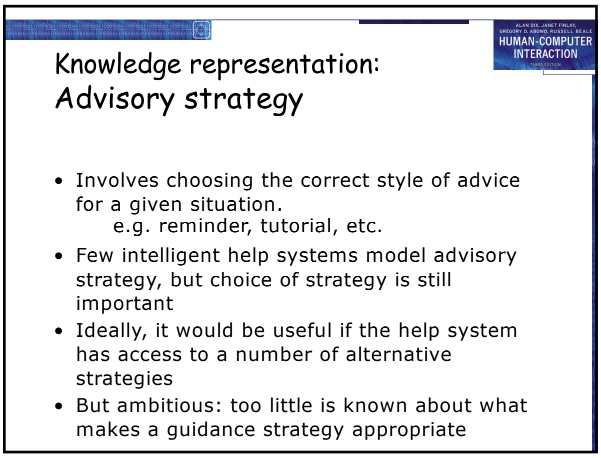
12



Knowledge representation: Domain and task modelling

- Covers
  - common errors and tasks
  - user’s current task or plan
  - motivation is that the user is engaged in a particular problem solving task and requires help at that level
- Usually involves analysis of command sequences
- Problems
  - representing tasks (many tasks)
  - interleaved tasks (many ways to reach a task)
  - inferring user intention (from command use) may not be straightforward, if the user approaches the task in a non-standard way

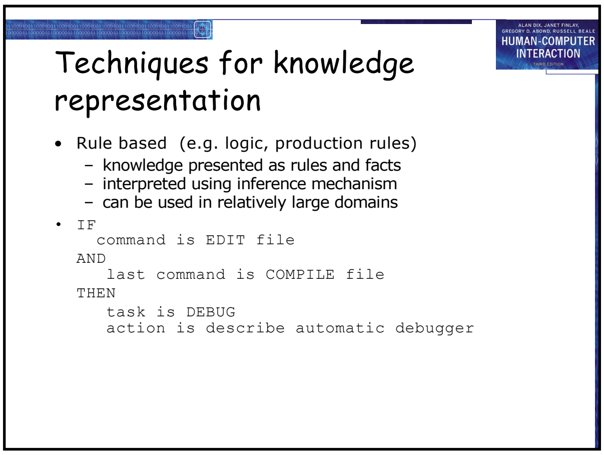
13



Knowledge representation: Advisory strategy

- Involves choosing the correct style of advice for a given situation.
  e.g. reminder, tutorial, etc.
- Few intelligent help systems model advisory strategy, but choice of strategy is still important
- Ideally, it would be useful if the help system has access to a number of alternative strategies
- But ambitious: too little is known about what makes a guidance strategy appropriate

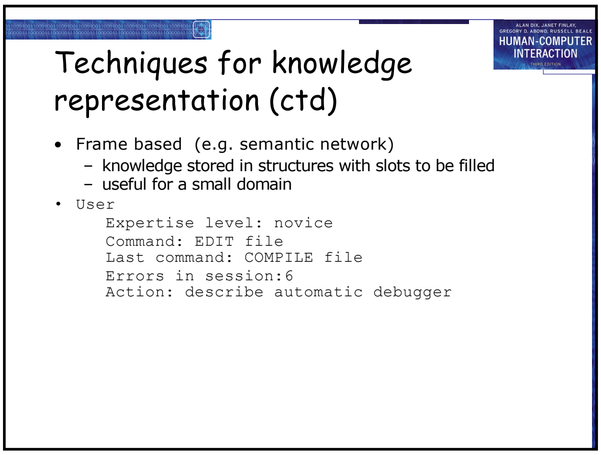
14



Techniques for knowledge representation

- Rule based (e.g. logic, production rules)
  - knowledge presented as rules and facts
  - interpreted using inference mechanism
  - can be used in relatively large domains
-
```
IF
  command is EDIT file
AND
   last command is COMPILE file
THEN
   task is DEBUG
   action is describe automatic debugger
```

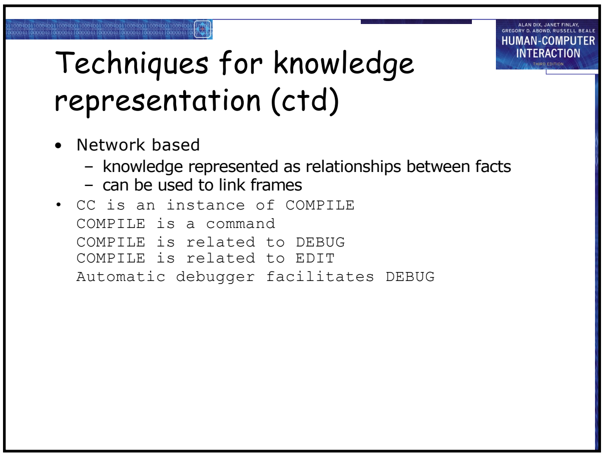
15



Techniques for knowledge representation (ctd)

- Frame based (e.g. semantic network)
  - knowledge stored in structures with slots to be filled
  - useful for a small domain
-
```
User
   Expertise level: novice
   Command: EDIT file
   Last command: COMPILE file
   Errors in session:6
   Action: describe automatic debugger
```

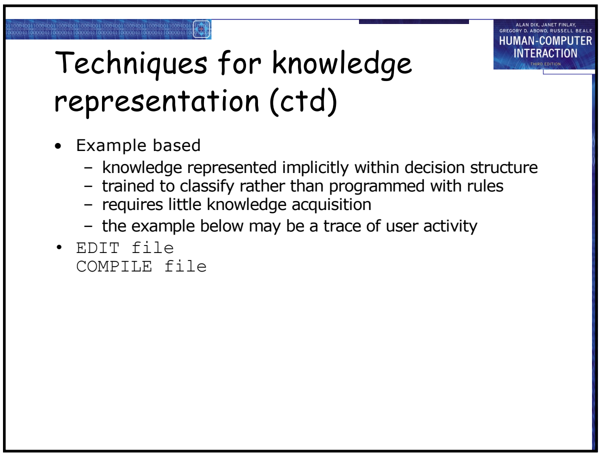
16



Techniques for knowledge representation (ctd)

- Network based
  - knowledge represented as relationships between facts
  - can be used to link frames
-
```
CC is an instance of COMPILE
COMPILE is a command
COMPILE is related to DEBUG
COMPILE is related to EDIT
Automatic debugger facilitates DEBUG
```

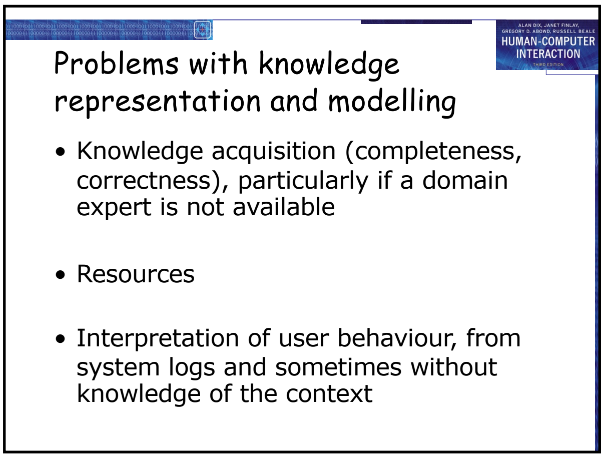
17



Techniques for knowledge representation (ctd)

- Example based
  - knowledge represented implicitly within decision structure
  - trained to classify rather than programmed with rules
  - requires little knowledge acquisition
  - the example below may be a trace of user activity
-
```
EDIT file
COMPILE file
```

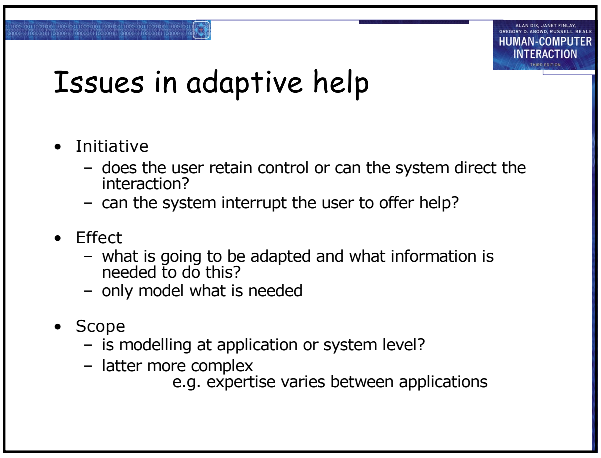
18



Problems with knowledge representation and modelling

- Knowledge acquisition (completeness, correctness), particularly if a domain expert is not available
- Resources
- Interpretation of user behaviour, from system logs and sometimes without knowledge of the context

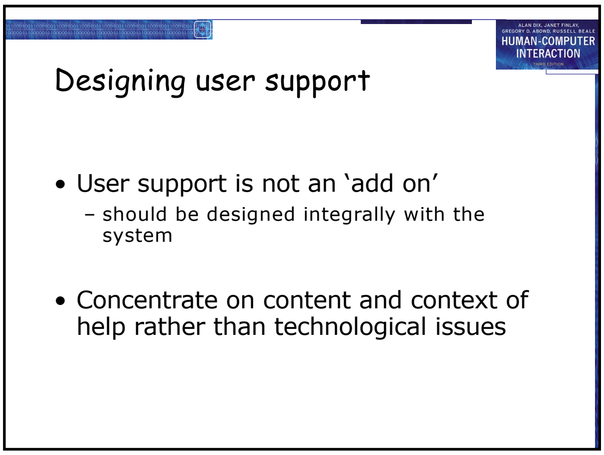
19



Issues in adaptive help

- Initiative
  - does the user retain control or can the system direct the interaction?
  - can the system interrupt the user to offer help?
- Effect
  - what is going to be adapted and what information is needed to do this?
  - only model what is needed
- Scope
  - is modelling at application or system level?
  - latter more complex
    e.g. expertise varies between applications

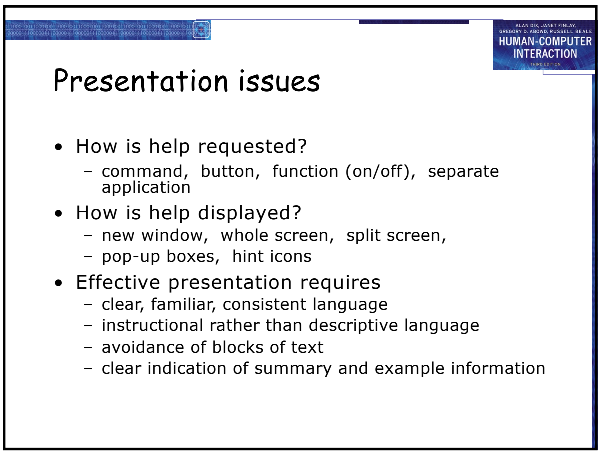
20



Designing user support

- User support is not an 'add on'
  - should be designed integrally with the system
- Concentrate on content and context of help rather than technological issues

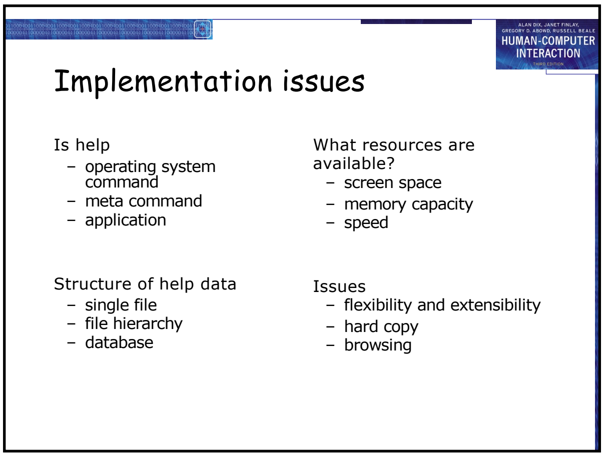
21



Presentation issues

- How is help requested?
  - command, button, function (on/off), separate application
- How is help displayed?
  - new window, whole screen, split screen,
  - pop-up boxes, hint icons
- Effective presentation requires
  - clear, familiar, consistent language
  - instructional rather than descriptive language
  - avoidance of blocks of text
  - clear indication of summary and example information

22



Implementation issues

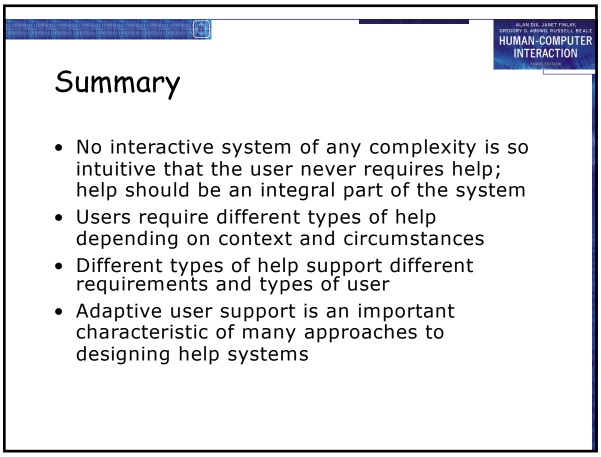
Is help
- operating system command
- meta command
- application

Structure of help data
- single file
- file hierarchy
- database

What resources are available?
- screen space
- memory capacity
- speed

Issues
- flexibility and extensibility
- hard copy
- browsing

23



Summary

- No interactive system of any complexity is so intuitive that the user never requires help; help should be an integral part of the system
- Users require different types of help depending on context and circumstances
- Different types of help support different requirements and types of user
- Adaptive user support is an important characteristic of many approaches to designing help systems

24