The context problem for software agents

Software agents provide assistance to users of interactive graphical interfaces

What should I do for the user?

How should I do it?

Agent needs to understand the intention of the user

Intention often can be inferred from context

Context reduces explicit input
Agent Personalities

Butler/Servant/Secretary
  • Agent executes commands, satisfies goals, anticipates needs

Teacher/Student, Master/Apprentice
  • Agent learns skills taught by user
The context problem for learning agents

Agent can only learn from concrete experience
Concrete experience needs to generalized
  Conservative: Stick close to experience
  Increased Accuracy
  Liberal: Try to do as much abstraction as possible
  Increased Applicability
Programming by Example [or "by Demonstration"]

Agent "watches what you do" in the interactive interface

Records sequence of operations, data involved in operation

Generalizes program so that you can use an analogous procedure in new examples
Strategies for generalization in Programming by Example

System *makes a guess* heuristically

Inferred from context, domain-dependent

System *asks user*

System may supply choices to give user context

System *receives advice* from user

Advice used as context for system's choices
The Data Description Problem

[Halber]

How should objects involved in examples be described?

Intentional vs. extensional descriptions
Hierarchical descriptions
Machine Learning: Version Spaces
Also: Action Description Problem, generally easier

Henry Lieberman • MIT Media Lab
The Critique Problem

Don't do *that* again!
What's *tha*?
Examples of PBE systems

Mondrian Graphical Editing
Grammex Text Recognition Agents
Agent Personalities

Butler/Servant/Secretary
- Agent executes commands, satisfies goals, anticipates needs

Teacher/Student, Master/Apprentice
- Agent learns skills taught by user
Mondrian:
An Instructible Graphical Editor

Mondrian=

• An object-oriented graphical editor +
• An agent that records user interface actions +
• Learning and generalization heuristics

Henry Lieberman • MIT Media Lab
Mondrian: An Instructible Graphical Editor
Web has renewed interest in text parsing technology

Much information on the Web and in desktop applications exists in “semi-structured” form

- Structured data embedded in unstructured data

Hiya --

Great news - we’re going to Moorea came through! Please call our travel agent at 555-9662 and give him your passport number. I’m going to be out of the office this afternoon, but you can leave a message with my secretary.  

E-Mail Address  
@apple.com

Also -- check out the lagoon views at http://www.com/islands/moorea/main.html -- really gorgeous looking scenery. Can’t wait to get there!

Later,  
John
Parsers are controlled by grammars

Grammar is a set of rules, each of which recognizes a class of text strings

Usually written in BNF or equivalent

Users have difficulty writing in a formal language

... but they DO understand the concepts behind recognition

“An e-mail address is [usually] a person’s name, followed by an “@”, followed by a host”
Solution: Define grammars by example!

Grammars are difficult because they are abstract

It's hard to understand what the effect of writing a rule will be in particular examples

Stresses short-term memory, reasoning

People are much better at dealing with concrete examples than abstractions

So, present concrete textual examples about the system how to interpret them
Grammex =
“Grammars by Example”

A “programming by example” interface for defining grammars

User supplies example of text to be recognized

System tries to parse text according to current grammar

User can interactively specialize and generalize interpretations of substrings

Grammex compiles a BNF-like grammar
Grammex rule windows

- Grammex breaks a string into substrings
- Popup menu of interpretations for each substring

Henry Lieberman • MIT Media Lab