Task Models

Presenter

Stephen Kimani
Universita' di Roma "La Sapienza"
DIS
Via Ariosto 25
00185 Rome
Italy

Web: http://www.dis.uniroma1.it/~kimani
E-mail: stephenkimani@gmail.com
Task Models

Roadmap

- Introduction to task analysis
- General method for task analysis
- Differences from other techniques
- Approaches to task analysis
- Task decomposition
- Knowledge-based techniques
- Entity/object-based techniques
- Purpose of task analysis
Introduction to Task Analysis

Task
This is an activity that has to be performed to achieve a goal.

Task analysis
It is the process of analyzing the way people perform tasks.
Task analysis refers to techniques that analyze:
- what people do
- what things they work with
- what they must know

For example: in order to clean the house,
One needs to do the following:
- get the vacuum cleaner out
- fix the appropriate attachments
- clean the rooms
- when the dust bag gets full, empty it
- put the vacuum cleaner and tools away

One works with: vacuum cleaner, the attachments, dust bags, etc
One must know about: vacuum cleaners, their attachments, dust bags, rooms, etc
General Method for Task Analysis

The general method for task analysis entails:
- observing the user’s behaviour
- collecting unstructured lists of words and actions
- organizing using notation or diagrams

Note that in task analysis, one should focus on
- the user’s objective observable behaviour (external actions)
and not on
- the user’s internal mental model
<table>
<thead>
<tr>
<th>Differences from Other Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Task analysis</strong></td>
</tr>
<tr>
<td>the user</td>
</tr>
<tr>
<td><strong>Task analysis</strong></td>
</tr>
<tr>
<td>external actions</td>
</tr>
</tbody>
</table>
Approaches to Task Analysis

- **1. Task decomposition**
  - Splitting task into (ordered) *subtasks*

- **2. Knowledge based techniques**
  - What the user *knows* about the task and how it is *organized*
  - The focus is on objects and actions
  - Taxonomies are created to represent levels of abstraction

- **3. Entity/object based analysis**
  - *Relationships* between objects, actions and the people who perform them
  - The focus is on *objects and actions*, and the *relationships* between those objects and actions
Task Decomposition

Aims

• Describe the actions people do
• Structure them within task subtask hierarchy
• Describe order of subtasks

There are several variants:
1. Hierarchical Task Analysis (HTA): the most common
2. ConcurTaskTrees (CTT), by Fabio Paternò (Pisa): uses temporal operators
Hierarchical Task Analysis (HTA)
- It is a task decomposition technique
- It has Hierarchy + Plans
  1. Hierarchy - hierarchy of tasks and subtasks
  2. Plans - the order of subtasks and the conditions under which they are performed (note that only the plans denote the order)

(see next two slides for illustration)
Textual HTA

Hierarchy

0. in order to clean the house
   1. get the vacuum cleaner out
   2. get the appropriate attachment
   3. clean the rooms
      3.1. clean the hall
      3.2. clean the living rooms
      3.3. clean the bedrooms
   4. empty the dust bag
   5. put vacuum cleaner and attachments away

Plans

Plan 0: do 1 - 2 - 3 - 5 in that order. when the dust bag gets full do 4
Plan 3: do any of 3.1, 3.2 or 3.3 in any order depending on which rooms need cleaning
Diagrammatic HTA

0. make a cup of tea

plan 0.
do 1
at the same time, if the pot is full 2
then 3 - 4
after four or five minutes do 5

1. boil water
2. empty pot
3. put tea leaves in pot
4. pour in boiling water
5. wait 4 or 5 minutes
6. pour tea

plan 1.
1.1 - 1.2 - 1.3
when kettle boils 1.4

1.1. fill kettle
1.2. put kettle on stove
1.3. wait for kettle to boil
1.4. turn off gas
Refining Preliminary HTA

Some heuristics for checking/improving an initial HTA

• Paired actions e.g., there is turn off but where is `turn on gas`?
• Restructure e.g., maybe too many tasks at the high level => generate task `make pot` to encompass say 3, 4, 5; maybe task 6 should have subtasks
• Balance e.g., are tasks on a particular of the same level of simplicity or abstraction, for instance is `pour tea` simpler than `make pot`?
• Generalize e.g., make one cup ... or more
Refined HTA

0. make cups of tea

plan 0.
do 1
at the same time, if the pot is full 2
then 3 – 4
after 4/5 minutes do 5

1. boil water
2. empty pot
3. make pot
4. wait 4 or 5 minutes
5. pour tea

plan 1.
1.1 – 1.2 – 1.3 – 1.4
when kettle boils 1.5

plan 3.
3.1 – 3.2 – 3.3

3.1. warm pot
3.2. put tea leaves in pot
3.3. pour in boiling water

5.1. put milk in cup
5.2. fill cup with tea
5.3. do sugar

plan 5.3.
5.3.1 — if wanted 5.3.2

5.1. ask guest about sugar
5.3.2. add sugar to taste

1.1. fill kettle
1.2. put kettle on stove
1.3. turn on and light gas
1.4. wait for kettle to boil
1.5. turn off gas
Types of Plans

- **fixed sequence** - 1.1 then 1.2 then 1.3
- **optional tasks** - if the pot is full 2
- **wait for events** - when kettle boils 1.4
- **cycles** - do 5.1 5.2 while there are still empty cups
- **time-sharing** - do 1; at the same time …
- **discretionary** - do any of 3.1, 3.2 or 3.3 in any order
- **mixtures** - most plans involve several of the above
Knowledge-based Techniques

- These techniques or analyses seek to describe/represent what the user knows about the task and how it is organized.

- The focus is on:
  1. objects - which are used during the task
  2. actions - which are performed

- The techniques use taxonomies
  1. Taxonomies are created to represent levels of abstraction
  2. The organization/grouping depends on purpose

The main aim of knowledge-based techniques is to:
- understand the knowledge needed to perform a task
  - help in production of training material
  - assess the amount of common knowledge between different tasks
Knowledge-based Examples

Example – Car control taxonomy
motor controls
  steering  steering wheel, indicators
  cruising  ignition, accelerator, clutch, gear stick
lighting
  external  headlights, hazard lights
  internal  courtesy light
wash/wipe
  wipers  front wipers, rear wipers
  washers  front washers, rear washers
heating  temperature control, air direction, fan, rear screen heater
parking  hand brake, door lock

Example – Kitchen taxonomy
kitchen items
  preparation  bowl, plate, chopping board
  cooking  frying pan, casserole, saucepan
dining  plate, soup bowl, casserole, glass
It is a knowledge-based technique.

It uses a special form of taxonomy called Task Descriptive Hierarchy (TDH).

There are three types of branch point in taxonomy:
1. XOR – normal taxonomy, object in one and only one branch
2. AND – object must be in both multiple classifications
3. OR – weakest case, object can be in one, many or none
Example of Task Descriptive Hierarchy (TDH)

Example

wash/wipe AND
  function XOR
    wipe  front wipers, rear wipers
    wash  front washers, rear washers
  position XOR
    front  front wipers, front washers
    rear  rear wipers, rear washers

wash/wipe AND - means that an object appears in both function and position,
function XOR - means that an object appears in either wipe or wash but not both,
position XOR - means that an object appears in either front or rear but not both.
Example of Task Descriptive Hierarchy (TDH)

Example

kitchen item AND
/___shape XOR
/   |___dished mixing bowl, casserole, saucepan,
/   |                   soup bowl, glass
/   |___flat plate, chopping board, frying pan
/___function OR
{___preparation mixing bowl, plate, chopping board
{___cooking frying pan, casserole, saucepan
{___dining XOR
   |___for food plate, soup bowl, casserole
   |___for drink glass

N.B. ‘/\{‘ used for branch types.
Entity/Object-based Analysis

- In entity/object based analysis, the focus is on
  1. objects
  2. actions
  3. relationships between those objects and actions

- The analysis is similar to object-oriented analysis, but
  1. includes non-computer entities
  2. emphasizes domain understanding not implementation

Example: ‘Vera’s Veggies’ – a market gardening firm
- owner/manager: Vera
- employees: Sam and Tony
- various tools including a spade and plough
- field
- glasshouse
- computer controlled irrigation system
Objects

Start with a list of objects and classify them:

- **Concrete objects:**
  simple things: spade, plough, glasshouse, field

- **Actors:**
  human actors: Vera, Sam, Tony, the customers
  non-human actors: computer controlled irrigation system

- **Composite objects:**
  sets: the team = Vera, Sam, Tony
Attributes

To the objects, add attributes:

• For instance, the computer controlled irrigation system may have the following attributes:
  ➢ status: on/off/faulty
  ➢ capacity: 100 litres/minute

• However, the emphasis is on object participation in tasks:
  ➢ keep only relevant attributes (e.g., status)
  ➢ no need for completeness, but convenient to be initially over-inclusive and drop unnecessary attributes later
Actions

List actions and associate with each:
- agent – who performs the actions
- patient – which is changed by the action
- instrument – used to perform action

Examples:
- Sam (agent) planted (action) the leeks (patient)
- Tony dug the field with the spade (instrument)
Objects and Actions

Object Sam human actor
   Actions:
   S1: drive tractor
   S2: dig the carrots

Object Vera human actor
   Actions: as worker
   V1: plant marrow seed
   V2: program irrigation controller
   Actions: as manager
   V3: tell Sam to dig the carrots

Object the men composite
   Comprises: Sam, Tony

Object glasshouse simple
   Attribute:
   humidity: 0-100%

Object Marrow simple
   Actions:
   M1: germinate
   M2: grow
   ...

Question: what is an event, and how are relationships represented?
1. **Requirement capture and system design**
   - lift focus from system to use
   - suggest candidates for automation
   - facilitate presentation and discussion in an interdisciplinary team
   - improve understanding of the application domain
   - uncover user’s conceptual model

1. **User interface design**
   - taxonomies suggest menu layout
   - object/action lists suggest user interface objects
   - task frequency guides default choices
   - task sequences guide dialogue design

1. **Supporting evaluation of the system**

2. **Documentation and training/teaching**
Task Models

Questions

Thanks