**Tasks and Task Analysis**

**Task:** the activity that has to be performed to achieve a goal
Tasks and Task Analysis

Task: the activity that has to be performed to achieve a goal

Task Analysis: the process of analysing the way people perform tasks:

• what people do
• what things they work with
• what they must know
Method for Task analysis

General Method

- observe the user’s behaviour
- collect unstructured lists of words and actions
- organise using notation or diagrams
Method for Task analysis

General Method

- observe the user’s behaviour
- collect unstructured lists of words and actions
- organise using notation or diagrams

Focus on the user’s **objective observable behaviour** rather than on the user’s **internal mental model**
Method for Task analysis

General Method

- observe the user’s behaviour
- collect unstructured lists of words and actions
- organise using notation or diagrams

Focus on the user’s objective observable behaviour rather than on the user’s internal mental model

However, it might involve building a conceptual model
Purpose of Task Analysis

• production of training material and documentation
Purpose of Task Analysis

- production of training material and documentation
- contribute to the design of a new system
  - building a conceptual model
  - generation of user interfaces
Approaches to Task Analysis

Three different approaches:
Approaches to Task Analysis

Three different approaches:

- task decomposition
- knowledge-based techniques
- entity-relationship-based analysis
Task Decomposition

- describe the actions people do
- structure them within task-subtask hierarchy
- describe order of subtasks
**Task Decomposition**

- describe the **actions** people do
- **structure** them within **task-subtask hierarchy**
- describe **order of subtasks**

**Hierarchical Task Analysis (HTA)**

- **text and diagrams** to show **hierarchy**
- **plans to describe** **order**
**HTA: Textual Notation**

Hierarchy description:

0. make a cup of tea
   1. boil water
   2. empty pot
   3. put tea leaves in pot
   4. pour in boiling water
   5. wait 5 minutes
   6. pour tea
HTA: Textual Notation

Hierarchy description:

0. make a cup of tea
   1. boil water
   2. empty pot
   3. put tea leaves in pot
   4. pour in boiling water
   5. wait 5 minutes
   6. pour tea

Plans

Plan 0. do 1
   at the same time, if pot is full do 2
   then do 3 – 4 – 5
   after 5 minutes do 6
Generating the Hierarchy

- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further
Generating the Hierarchy

- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further

How to know when to stop?
Generating the Hierarchy

- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further

How to know when to stop?

Stopping rules:
- **Simplicity:** Is the task simple enough?
- **Purpose:** Is the task relevant?
- **Motor Action:** lowest sensible level
HTA: Diagrammatic Notation

0. make a cup of tea

plan 0.
do 1
at the same time, if pot is full do 2 then do 3 – 4 – 5 after 5 minutes do 6

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

0. make a cup of tea
   plan 0.
   do 1
   at the same time, if pot is full do 2
   then do 3 – 4 – 5
   after 5 minutes do 6

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

0. make a cup of tea
   plan 0.
   do 1 at the same time, if pot is full do 2
   then do 3 – 4 – 5
   after 5 minutes do 6

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

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

1.1. fill kettle
1.2. put kettle on stove
1.3. wait for kettle to boil
1.4. turn off gas
HTA: Domain Expert

0. make a cup of tea
   plan 0.
   do 1
   at the same time, if pot is full do 2
   then do 3 – 4 – 5
   after 5 minutes do 6

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

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

Looking for errors
Describe the step in the task hierarchy to a domain expert
**HTA: Domain Expert**

0. make a cup of tea

plan 0.
- do 1 at the same time, if pot is full do 2
- then do 3 – 4 – 5
- after 5 minutes do 6

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

plan 1.
- do 1.1 – 1.2 – 1.3
- when kettle boils do 1.5

**Looking for errors**

Describe the step in the task hierarchy to a domain expert

**We forgot to warm the pot**
**HTA: Domain Expert**

0. **make a cup of tea**
   - plan 0.
     - do 1
     - at the same time, if pot is full do 2
     - then do 3 – 4 – 5
     - after 5 minutes do 6

1. **boil water**

2. **empty pot**

3. **put tea leaves in pot**

4. **pour in boiling water**

5. **wait 5 minutes**

6. **pour tea**

plan 1.
- do 1.1 – 1.2 – 1.3
- when kettle boils do 1.5

Looking for errors
Describe the step in the task hierarchy to a domain expert

*We forgot to warm the pot*
HTA: Omissions

0. make a cup of tea

- plan 0.
  - do 1
  - at the same time, if pot is full do 2
  - then do 3 – 4 – 5 – 6
  - after 5 minutes do 7

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

- plan 1.
  - do 1.1 – 1.2 – 1.3
  - when kettle boils do 1.5

1.1. fill kettle
1.2. put kettle on stove
1.3. wait for kettle to boil
1.4. turn off gas
**HTA: Omissions**

0. make a cup of tea

plan 0.
- do 1 at the same time, if pot is full do 2 then do 3 – 4 – 5 – 6 after 5 minutes do 7

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

plan 1.
- do 1.1 – 1.2 – 1.3 when kettle boils do 1.5

**Omissions?**
HTA: Omissions

0. make a cup of tea

plan 0.
do 1
at the same time, if pot is full do 2
then do 3 – 4 – 5 – 6
after 5 minutes do 7

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

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

Omissions?
HTA: Omissions

0. make a cup of tea

plan 0.
do 1 at the same time, if pot is full do 2 then do 3 – 4 – 5 – 6 after 5 minutes do 7

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

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

Omissions?
Where do we turn the gas on?

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

plan 0.
do 1
at the same time, if pot is full do 2
then do 3 – 4 – 5 – 6
after 5 minutes do 7

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

Omissions?
Where do we turn the gas on?
Maybe implicit in
**HTA: Omissions**

0. make a cup of tea
   - plan 0.
     - do 1 at the same time, if pot is full do 2
     - then do 3 – 4 – 5 – 6
     - after 5 minutes do 7

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

plan 1.
- do 1.1 – 1.2 – 1.3 when kettle boils do 1.5

**Omissions?**

*Where do we turn the gas on? We make it explicit here.*

1.1. fill kettle
1.2. put kettle on stove
1.3. wait for kettle to boil
1.4. turn off gas
**HTA: Umbalanced Hierarchy**

0. make a cup of tea

plan 0.
do 1 at the same time, if pot is full do 2 then do 3 – 4 – 5 – 6 after 5 minutes do 7

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

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

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
**HTA: Unbalanced Hierarchy**

0. make a cup of tea
   
   plan 0.
   do 1
   at the same time, if pot is full do 2
   then do 3 – 4 – 5 – 6
   after 5 minutes do 7

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

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

Why is the hierarchy unbalanced?
HTA: Unbalanced Hierarchy

0. make a cup of tea
   plan 0.
   do 1
   at the same time, if pot is full do 2
   then do 3 – 4 – 5 – 6
   after 5 minutes do 7

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

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

Why is the hierarchy unbalanced?
Maybe too many details at the high level
HTA: Unbalanced Hierarchy

Why is the hierarchy unbalanced? Maybe too many details at the high level.
**HTA: Unbalanced Hierarchy**

### Tasks

0. **make a cup of tea**
   - plan 0.
   - do 1 at the same time, if pot is full do 2 then do 3 – 4 – 5 – 6 after 5 minutes do 7

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

Plan 1.
- do 1.1 – 1.2 – 1.3 when kettle boils do 1.5

### Why is the hierarchy unbalanced?
Maybe too many details at the high level

We add new make pot node which encompass tasks 3, 4, 5. Why not 2 and 6?
HTA: Further Decompositions

<table>
<thead>
<tr>
<th>Plan</th>
<th>Task Decomposition</th>
<th>Task Knowledge</th>
<th>Task ER</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>0. make a cup of tea</td>
<td>do 0. at the same time, if pot is full do 2 then do 3 – 4 after 5 minutes do 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. boil water</td>
<td>plan 1. do 1.1 – 1.2 – 1.3 when kettle boils do 1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. empty pot</td>
<td>plan 3. do 3.1 – 3.2 – 3.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. make pot</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. wait 5 minutes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. pour tea</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HTA: Further Decompositions

0. make a cup of tea

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

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

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

plan 3.
do 3.1 – 3.2 – 3.3

A. Cerone, UNU-IIST – p.14/38
HTA: Further Decompositions

0. make a cup of tea
  plan 0.
  do 1 at the same time, if pot is full do 2
  then do 3 – 4
  after 5 minutes do 5

1. boil water
  plan 1.
  do 1.1 – 1.2 – 1.3
  when kettle boils do 1.5

2. empty pot

3. make pot
  plan 3.
  do 3.1 – 3.2 – 3.3

4. wait 5 minutes

5. pour tea
  plan 5.
  do 5.1 – 5.2
  if desired do 5.3

5.1. put milk in cup

5.2. fill cup with tea

5.3. add sugar

3.1. warm pot

3.2. put tea leaves in pot

3.3. pour in boiling water

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
HTA: Iteration

0. make a cup of tea
  
1. boil water
2. empty pot
3. make pot
4. wait 5 minutes
5. pour tea

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

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

plan 3.
do 3.1 – 3.2 – 3.3

plan 5.
do 5.1 – 5.2
if desired do 5.3

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

2.1. fill kettle
2.2. put kettle on stove
2.3. turn on and light gas
2.4. wait for kettle to boil
2.5. turn off gas

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. add sugar

A. Cerone, UNU-IIST – p.15/38
HTA: Iteration

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

1. boil water
   plan 1.
   do 1.1 – 1.2 – 1.3
   when kettle boils do 1.5

2. empty pot

3. make pot
   plan 3.
   do 3.1 – 3.2 – 3.3

4. wait 5 minutes

5. pour tea
   plan 6.
   do 5.1 – 5.2
   if desired do 5.3

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. add sugar

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

plan 5.

A. Cerone, UNU-IIST – p.15/38
HTA: Iteration

0. make cups of tea

1. boil water

2. empty pot

3. make pot

4. wait 5 minutes

5. pour tea

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

- plan 1.
  - do 1.1 – 1.2 – 1.3
    - when kettle boils do 1.5

- plan 3.
  - do 3.1 – 3.2 – 3.3

- plan 5.
  - 5.1
  - 5.2
  - 5.3
    - add sugar

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

2.1. put kettle on stove

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. add sugar

for each guest do 5.3

empty cups?

NO

YES

for each guest

do 5.3

6. empty cups?

A. Cerone, UNU-IIST – p.15/38
HTA: Final Decomposition

0. make cups of tea

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

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

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

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

for each guest
plan 5. do 5.3

empty cups?

YES

no

A. Cerone, UNU-IIST – p.16/38
HTA: Fixed Sequence

0. make cups of tea

1. boil water
   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

2. empty pot
   2.1. fill kettle
   2.2. put kettle on stove
   2.3. turn on and light gas
   2.4. wait for kettle to boil
   2.5. turn off gas

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

4. wait 5 minutes

5. pour tea
   5.1. put milk in cup
   5.2. fill cup with tea
   5.3. do sugar
      5.3.1. ask guest about sugar
      5.3.2. add sugar to taste

For each guest:
   5.3

Plan 0:
- Do 1 at the same time, if pot is full do 2
- Then do 3 - 4
- After 5 minutes do 5

Plan 1:
- Do 1.1 - 1.2 - 1.3 - 1.4
- When kettle boils do 1.5

Plan 3:
- Do 3.1 - 3.2 - 3.3

Plan 5:
- Do 5.1
- Do 5.2
- Empty cups?
  - YES
  - NO

If wanted 5.3.2:
- Do 5.3.1

A. Cerone, UNU-IIST – p.17/38
HTA: Optional Tasks

0. make cups of tea

1. boil water
2. empty pot
3. make pot
4. wait 5 minutes
5. pour tea
5.1. put milk in cup
5.2. fill cup with tea
5.3. do sugar
5.3.1. 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

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

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

plan 3.
do 3.1 – 3.2 – 3.3

plan 5.
do 5.1 – 5.2

for each guest do 5.3

6 empty cups?

YES

for each guest do 5.3

plan 5.3.
do 5.3.1 – if wanted 5.3.2

A. Cerone, UNU-IIST – p.18/38
HTA: Waiting for Events

Plan 0:
- **0.** make cups of tea
  - At the same time, if pot is full do 2
  - Then do 3 – 4
  - After 5 minutes do 5

Plan 1:
- **1.** boil water
  - **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
- **2.** empty pot
- **3.** make pot
  - **3.1.** warm pot
  - **3.2.** put tea leaves in pot
  - **3.3.** pour in boiling water

Plan 3:
- **3.1.** put milk in cup
- **3.2.** fill cup with tea
- **3.3.** do sugar
  - **3.3.1.** ask guest about sugar
  - **3.3.2.** add sugar to taste

Plan 5:
- **5.** pour tea
  - **5.1.** put milk in cup
  - **5.2.** fill cup with tea
  - **5.3.** do sugar
    - **5.3.1.** ask guest about sugar
    - **5.3.2.** add sugar to taste

A. Cerone, UNU-IIST – p.19/38
HTA: Cycles

0. make cups of tea

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

Plan 1.
- do 1.1 – 1.2 – 1.3 – 1.4 when kettle boils do 1.5

Plan 3.
- do 3.1 – 3.2 – 3.3

Plan 5.
- do 5.1
- do 5.2
  - empty cups?
    - NO
    - YES
  - for each guest do 5.3

Plan 5.3.
- do 5.3.1 – if wanted 5.3.2

1. boil water

1.1. fill kettle

2. empty pot

2.1. put kettle on stove

3. make pot

3.1. warm pot

3.2. put tea leaves in pot

3.3. pour in boiling water

4. wait 5 minutes

4.1. turn on and light gas

5. pour tea

5.1. put milk in cup

5.2. fill cup with tea

5.3. do sugar

5.3.1. ask guest about sugar

5.3.2. add sugar to taste

6. empty cups?
**HTA: Time-sharing**

0. **make cups of tea**
   - plan 0.
     - do 1 at the same time, if pot is full do 2
     - then do 3 – 4
     - after 5 minutes do 5

1. **boil water**
   - plan 1.
     - do 1.1 – 1.2 – 1.3 – 1.4
     - when kettle boils do 1.5

2. **empty pot**

3. **make pot**
   - plan 3.
     - do 3.1 – 3.2 – 3.3

4. **wait 5 minutes**

5. **pour tea**
   - plan 5.
     - 5.1
     - 5.2
     - empty cups?
       - NO
       - YES

5.1. **put milk in cup**

5.2. **fill cup with tea**

5.3. **do sugar**
   - plan 5.3.
     - do 5.3.1 – if wanted 5.3.2

5.3.1. **ask guest about sugar**

5.3.2. **add sugar to taste**

for each guest do 5.3

- 6

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**

---

A. Cerone, UNU-IIST – p.21/38
HTA: Mixtures

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

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

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

plan 3.
do 3.1 – 3.2 – 3.3

plan 5.
do 5.1
for each guest
do 5.3

5.1. put milk
5.2. fill cup
5.3. do sugar
5.1. put milk
5.2. fill cup
5.3. do sugar

5.1. put milk
5.2. fill cup
5.3. do sugar

5.3.1. ask guest
5.3.2. add sugar

for each guest
do 5.3

5.3.1. ask guest
5.3.2. add sugar

A. Cerone, UNU-IIST – p.22/38
Decomposition Heuristics

- **paired actions**
  e.g., turn on and turn on gas

- **restructure/balance** e.g., generate make pot and decompose pour tea

- **generalise**
  e.g., from make a cup of tea to make cups of tea
Knowledge-based Analysis

- list **objects** used in tasks
- list **actions** performed
- build **taxonomies** of them
Knowledge-based Analysis

- list objects used in tasks
- list actions performed
- build taxonomies of them

Aim

- to understand knowledge needed to perform a task
- help in production of teaching material
- assess the amount of common knowledge between different tasks
**Knowledge-based Analysis**

- list **objects** used in tasks
- list **actions** performed
- build **taxonomies** of them

**Aim**

- to understand **knowledge** needed to perform a task
- $\implies$ help in production of **teaching material**
- $\implies$ assess the amount of **common knowledge** between different tasks

**Technique:** Task Analysis for Knowledge Description — TAKD
Example: Kitchen Items

kitchen items
preparation
bowl, plate, chopping board
cooking
frying pan, casserole, saucepan
dining
plate, soup bowl, casserole, glass
**TDH notation**

**TDH — Task Descriptive Hierarchy**

- kitchen item OR
  - preparation
    - bowl, plate, chopping board
  - cooking
    - frying pan, casserole, saucepan
  - dining
    - plate, soup bowl, casserole, glass
**TDH notation**

TDH — Task Descriptive Hierarchy

kitchen item OR

{___ preparation
{    bowl, plate, chopping board
{___ cooking
{    frying pan, casserole, saucepan
{___ dining
    plate, soup bowl, casserole, glass

*Uniqueness Rule:*
A complete TDH can distinguish any two specific objects
TDH notation

TDH — Task Descriptive Hierarchy

kitchen item OR

{___ preparation
{ bowl, plate, chopping board
{___ cooking
{ frying pan, casserole, saucepan
{___ dining
plate, soup bowl, casserole, glass

Uniqueness Rule:
A complete TDH can distinguish any two specific objects
TDH: Branching Type

kitchen item AND
/ ___ function OR
/ / ___ preparation
/ / { ___ bowl, plate, chopping board
/ / ___ cooking
/ / ___ frying pan, casserole, saucepan
/ / ___ dining XOR
/ / ___ for food
/ / ___ plate, soup bowl, casserole
/ / ___ for drink
/ / ___ glass
/ ___ shape XOR
...

A. Cerone, UNU-IIST – p.27/38
TDH: Branching Type

kitchen item AND
/___ function OR
/  __ preparation
/    bowl, plate, chopping board
/    cooking
/    frying pan, casserole, saucepan
/    dining XOR
/      for food
/      plate, soup bowl, casserole
/      for drink
/      glass
/___ shape XOR
...

kitchen item/function{dining(for food)/shape(dished)}/

KRG — Knowledge Representation Grammar
TDH: Branching Type

kitchen item AND
/___ function OR
/___ preparation
/   {___ bowl, plate, chopping board
/   {___ cooking
/   {___ frying pan, casserole, saucepan
/   {___ dining XOR
/   |___ for food
/   |___ plate, soup bowl, casserole
/   |___ for drink
/___ shape XOR
...

kitchen item/function{dining(for drink)/shape(dished)}/

KRG — Knowledge Representation Grammar
TDH: Branching Type

kitchen item AND
/___ function OR
/    {___ preparation
/    {___ bowl, plate, chopping board
/    {___ cooking
/    {___ frying pan, casserole, saucepan
/    {___ dining XOR
/    {___ for food
/    {___ plate, soup bowl, casserole
/    {___ for drink
/    {___ glass
/___ shape XOR
...

kitchen item/function{preparation,dining(for food)/shape(flat)}/

KRG — Knowledge Representation Grammar
Taxonomy of Actions

kitchen job OR

{___ preparation
{___ beating, mixing
{___ cooking
{___ frying, boiling, baking
{___ dining
pouring, eating, drinking
**ER-based techniques**

Entity-Relationship Based Techniques

- list **objects** used in tasks
- list **actions** performed
- define **relationships** between object and actions
**ER-based techniques**

Entity-Relationship Based Techniques

- list **objects** used in tasks
- list **actions** performed
- define **relationships** between object and actions
- similar to techniques used in **database** and **OO**
- but includes **non-computer entities**
- emphasis on **domain understanding** rather than implementation
Example: Vera’s Veggies

- Vera’s Veggies a market gardening firm
- owner/manager: Vera
- employes: Sam and Tony
- tools include a tractor Fergie
- two fields and a glasshouse
- new computer controlled irrigation system
Object Classification

- concrete objects
  simple things: spade, plough, glasshouse
Object Classification

- **concrete objects**
  simple things: spade, plough, glasshouse

- **actors**
  human: Vera, Sam, Tony, the customers
Object Classification

- **concrete objects**
  simple things: spade, plough, glasshouse

- **actors**
  human: Vera, Sam, Tony, the customers
  non-human: irrigation system
Object Classification

- **concrete objects**
  simple things: spade, plough, glasshouse

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

- **composite objects**
  sets: the team = Vera, Sam, Tony
Object Classification

- **concrete objects**
  - simple things: spade, plough, glasshouse

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

- **composite objects**
  - sets: the team = Vera, Sam, Tony
  - tuples: tractor = <Fergie, plough>
Attributes

An irrigation pump may have:

- **status**: on/off/faulty
- **capacity**: 100 litres/minute
Attributes

An irrigation pump may have:

- **status**: on/off/faulty
- **capacity**: 100 litres/minute

However, emphasis on object participation in tasks:

- keep only **relevant attributes** (e.g., status)
- **no need for completeness**, but convenient to be initially **overinclusive** and drop unnecessary attributes **later**
Actions

Agent performs Action to change Patient
Actions

Agent performs Action to change Patient

Sam planted the leeks
Actions

Agent performs Action to change Patient using Instrument

Sam planted the leeks
## Actions

<table>
<thead>
<tr>
<th>Agent</th>
<th>performs</th>
<th>Action to change Patient using Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>planted</td>
<td>the leeks</td>
</tr>
<tr>
<td>Tony</td>
<td>dug</td>
<td>the field</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with the spade</td>
</tr>
</tbody>
</table>
# Actions

<table>
<thead>
<tr>
<th>Agent</th>
<th>performs</th>
<th>Action</th>
<th>to change</th>
<th>Patient</th>
<th>using</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>planted</td>
<td>the leeks</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tony</td>
<td>dug</td>
<td>the field</td>
<td>with</td>
<td></td>
<td></td>
<td>the spade</td>
</tr>
<tr>
<td>Vera</td>
<td>turns on</td>
<td>the irrigation system</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agent</td>
<td>Action</td>
<td>Patient</td>
<td>using</td>
<td>Instrument</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td>-------------</td>
<td>----------</td>
<td>---------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sam</td>
<td>planted leeks</td>
<td>leeks</td>
<td></td>
<td>Instrument</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tony</td>
<td>dug field</td>
<td>field</td>
<td>with</td>
<td>spade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vera</td>
<td>turns on system</td>
<td>irrigation</td>
<td></td>
<td>system</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Irrigation system is turned on (control) (automatically)
## Actions

<table>
<thead>
<tr>
<th>Agent</th>
<th>Action</th>
<th>Patient</th>
<th>Instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
<td>planted</td>
<td>the leeks</td>
<td>Instrument</td>
</tr>
<tr>
<td>Tony</td>
<td>dug</td>
<td>the field</td>
<td>the spade</td>
</tr>
<tr>
<td>Vera</td>
<td>turns on</td>
<td>the irrigation system</td>
<td>(control)</td>
</tr>
<tr>
<td></td>
<td>is turned on</td>
<td>(automatically)</td>
<td>(indirect agent)</td>
</tr>
<tr>
<td>Vera</td>
<td>programmed</td>
<td>the irrigation system</td>
<td>...</td>
</tr>
</tbody>
</table>

*Vera programmed the irrigation system...*
Actions

Agent | performs | Action | to change | Patient | using | Instrument
--- | --- | --- | --- | --- | --- | ---
Sam | Sam | planted | the leeks | | | |
Tony | Tony | dug | the field | with | the spade | |
Vera | Vera | turns on | the irrigation system | | | |
irrigation system (control) | | is turned on (automatically) | | | | |
Vera | Vera | programmed | the irrigation system | | | |
(indirect agent) | | | | | | |
Vera | Vera | told | Sam | to | | |
(message) | | | | | | |
A. Cerone, UNU-IIST – p.33/38
**Actions**

**Agent** performs **Action** to change **Patient** using **Instrument**

- Sam planted the leeks
- Tony dug the field with the spade
- Vera turns on the irrigation system

**irrigation system** is turned on (automatically)

- Vera programmed the irrigation system
  (indirect agent)

- Vera told Sam to ...
  (message)

**Vera as worker** ...
**Vera as manager** ...

(agents may act in several roles)
Object and Actions

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

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

Object Vera human actor | the proprietor
Actions: as worker
  V1: plant marrow seed
  V2: programme irrigation controller
Actions: as manager
  V3: tell sam to dig the carrots
Object and Actions

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

Object Vera human actor | the proprietor
Actions: as worker
  V1: plant marrow seed
  V2: programme irrigation controller
Actions: as manager
  V3: tell sam to dig the carrots

Object the men composite
Comprises: Sam, Tony
Object and Actions

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

Object Vera human actor | the proprietor
Actions: as worker
   V1: plant marrow seed
   V2: programme 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 and Actions**

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

**Object Vera** human actor | the proprietor  
**Actions:** as worker  
- V1: plant marrow seed  
- V2: programme 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 Irrigation Controller** non-human  
**Actions:**  
- IC1: turn on Pump 1  
- IC2: turn on Pump 2  
- IC3: turn on Pump 3
Events

- performing of an action

Sam dug the carrots
Events

- performing of an action
  Sam dug the carrots

- spontaneous events
  the marrow seeds germinated
Events

• performing of an action
  Sam dug the carrots

• spontaneous events
  the marrow seeds germinated
  the humidity drops below 25%
Events

- **performing of an action**
  - Sam dug the carrots

- **spontaneous events**
  - the marrow seeds germinated
  - the humidity drops below 25%

- **timed events**
  - at midnight
Relationships

Object Marrow simple
Actions:
- M1: germinate
- M2: grow
Relationships

Object Marrow simple

Actions:
M1: germinate
M2: grow

Events
Ev1: humidity drops below 25%
Ev2: midnight
Relationships

Object  Marrow  simple
Actions:
  M1: germinate
  M2: grow
Events
  Ev1: humidity drops below 25%
  Ev2: midnight
Relations  object-object
location ( Pump 1, glasshouse )
**Relationships**

**Object** Marrow simple

**Actions:**
- M1: germinate
- M2: grow

**Events**
- Ev1: humidity drops below 25%
- Ev2: midnight

**Relations** object-object
- location (Pump 1, glasshouse)

**Relation** action-object
- patient (V3, Sam)
  - Vera tells Sam to dig
- patient (S2, the carrots)
  - Sam digs the carrots...
- instrument (S2, spade)
  - ... with the spade
Relations action-event

Relations action-event
before ( V1, M1 )
   — the marrow must be sown before it can germinate
before ( M1, M2 )
   — the marrow must germinate before it can grow

V1: plant marrow seed
M1: germinate
M2: grow
Relations action-event

Relations action-event
before ( V1, M1 )
   — the marrow must be sown before it can germinate
before ( M1, M2 )
   — the marrow must germinate before it can grow
triggers ( Ev2, IC1 )
   — when it is midnight the control turns on Pump 1

Ev1: humidity drops below 25%
IC1: turn on Pump 1
Relations action-event

Relations action-event
before ( V1, M1 )
— the marrow must be sown before it can germinate
before ( M1, M2 )
— the marrow must germinate before it can grow
triggers ( Ev2, IC1 )
— when it is midnight the control turns on Pump 1
causes ( V2, IC1 )
— the controller turns on Pump 1
because Vera programmed it

V2: programme irrigation controller
IC1: turn on Pump 1
Relations action-event

Relations action-event
before ( V1, M1 )
— the marrow must be sown before it can germinate
before ( M1, M2 )
— the marrow must germinate before it can grow
triggers ( Ev2, IC1 )
— when it is midnight the control turns on Pump 1
causes ( V2, IC1 )
— the controller turns on Pump1 because Vera programmed it
causes ( V3, S2 )
— Sam digs the carrots because Vera told him to do so

V3: tell sam to dig the carrots
S2: dig carrots
Relations action-event

Relations action-event
before ( V1, M1 )
— the marrow must be sown before it can germinate
before ( M1, M2 )
— the marrow must germinate before it can grow
triggers ( Ev2, IC1 )
— when it is midnight the control turns on Pump 1
causes ( V2, IC1 )
— the controller turns on Pump1
because Vera programmed it
causes ( V3, S2 )
— Sam digs the carrots
because Vera told him to do so

Ordering of events better described using HTA either to analyse order of subtasks and actions annotated with objects or to represent the life-cycle of a specific object
Uses of Task Analysis

Purposes:

- requirement capture and system design
  - lift focus from system to use
  - suggest candidates for automation
  - uncover user’s conceptual model
Uses of Task Analysis

Purposes:

• requirement capture and system design
  • lift focus from system to use
  • suggest candidates for automation
  • uncover user’s conceptual model

• interface design
  • taxonomies suggest menu layout
  • object/action lists suggest interface objects
  • task frequency guides default choices
  • task sequences guide dialogue design
Uses of Task Analysis

Purposes:

- requirement capture and system design
  - lift focus from system to use
  - suggest candidates for automation
  - uncover user’s conceptual model

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

- documentation and teaching