Help, It looks Confusing:

GUI Task Automation Through Demonstration and Follow-up Questions

Thanapong Intharah,
Daniyar Turmukhambetov
Gabriel J. Brostow

University College London

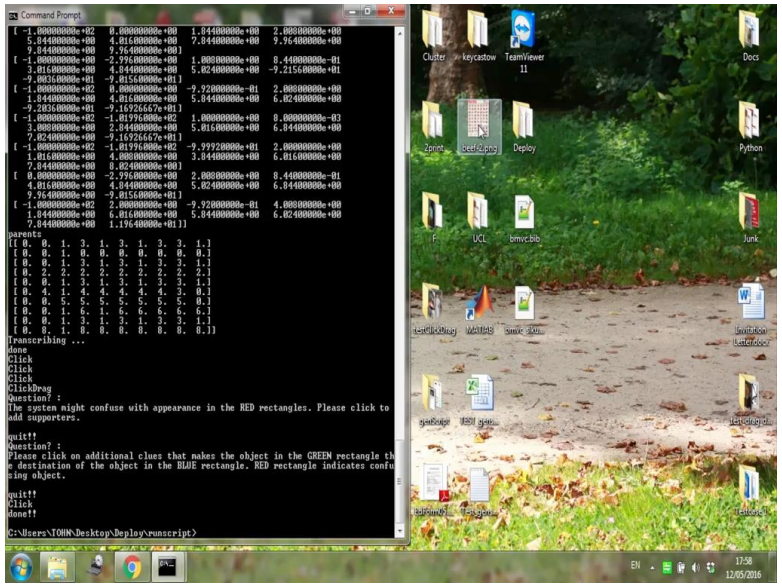
Target Users:

People Who Want To …

Target Users: People Who Want To …

(Linear) Have a macro/script where one click does many actions.

Example: switch everything to high contrast mode.



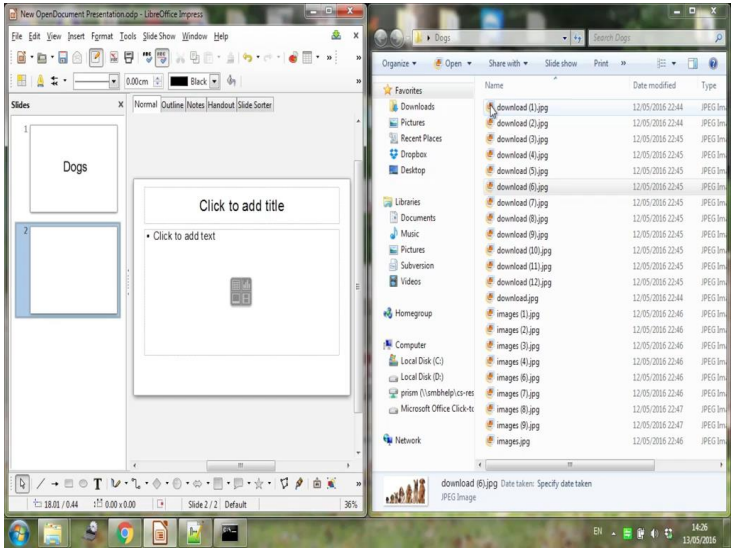
Target Users: People Who Want To …

(Linear) Have a macro/script where one click does many actions.

Example: switch everything to high contrast mode.

(Loop) Do iterative tasks, automatically.

Example: make each photo in this folder into a separate slide in PowerPoint.



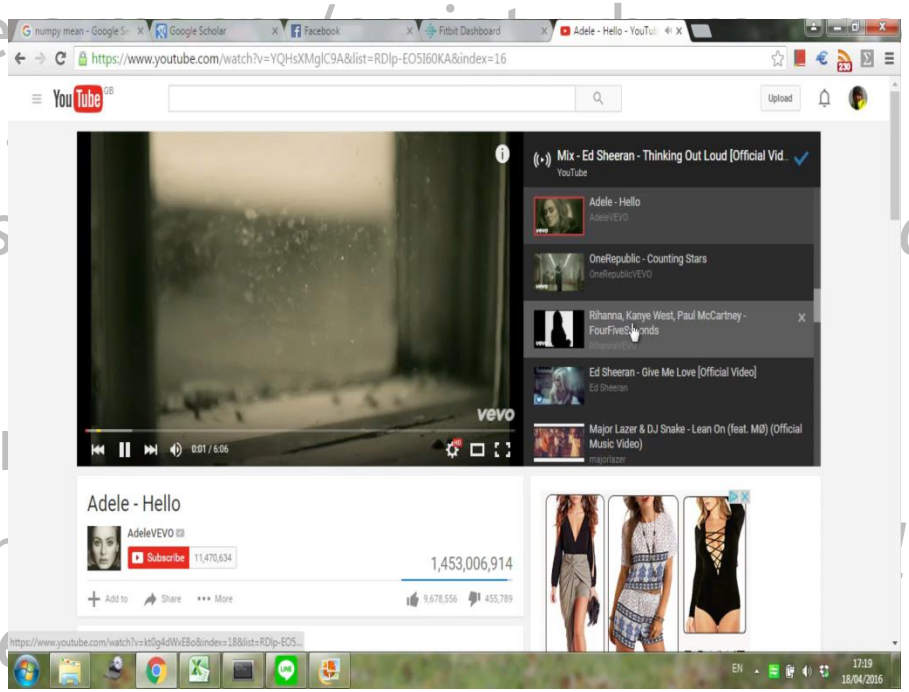
Target Users: People Who Want To …

(Linear) Have… click does many actions…

Example: s… ontrast mode.

(Loop) Do iter…

Example: r… lder into a separate slid…



(Monitoring) React the same way each time …

Example: click the "skip ad" during a youtube video.

Before HILC

Before HILC

- Nearest neighbors : Programming by Demonstration
  - Sikuli, Sikuli Slides *(Yeh et al 2009)*.
  - Koala *(Little et al 2007)*, CoScripter *(Leshed et al 2008)*.
  - More systems can be found in “Watch what I do” *(Cypher and Halbert 1993)*.

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli			
Sikuli Slides			
Koala, CoScriptor			
Sheepdog, Familiar			
HILC			

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli			
Sikuli Slides			
Koala, CoScriptor			
Sheepdog, Familiar			
HILC			

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli	✓	✗	✓
Sikuli Slides	✓	✓	✗
Koala, CoScriptor	✗	✓	✓
Sheepdog, Familiar	✗	✓	✓
HILC	✓	✓	✓

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli			
Sikuli Slides			
Koala, CoScriptor			
Sheepdog, Familiar			
HILC			

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli			
Sikuli Slides			
Koala, CoScriptor			
Sheepdog, Familiar			
HILC			

Feature Comparison

	Domain Independent	No Programming Skill Required	Deal with Non-linear Tasks
Sikuli			
Sikuli Slides			
Koala, CoScriptor			
Sheepdog, Familiar			
HILC			

Before HILC

- Nearest neighbours : Programming by Demonstration
  - Sikuli, Sikuli Slides *(Yeh et al 2009)*.
  - Koala *(Little et al 2007)*, CoScripter *(Leshed et al 2008)*.
  - More systems can be found in "Watch what I do" *(Cypher and Halbert 1993)*.

- Differences :
  - HILC is software-agnostic;

  no need for Accessibility API's and works across applications.

  - Inputs users feed to HILC resemble how a human

  teaches another to complete a task.

  - Able to deal with non-linear task.

From User's Perspective

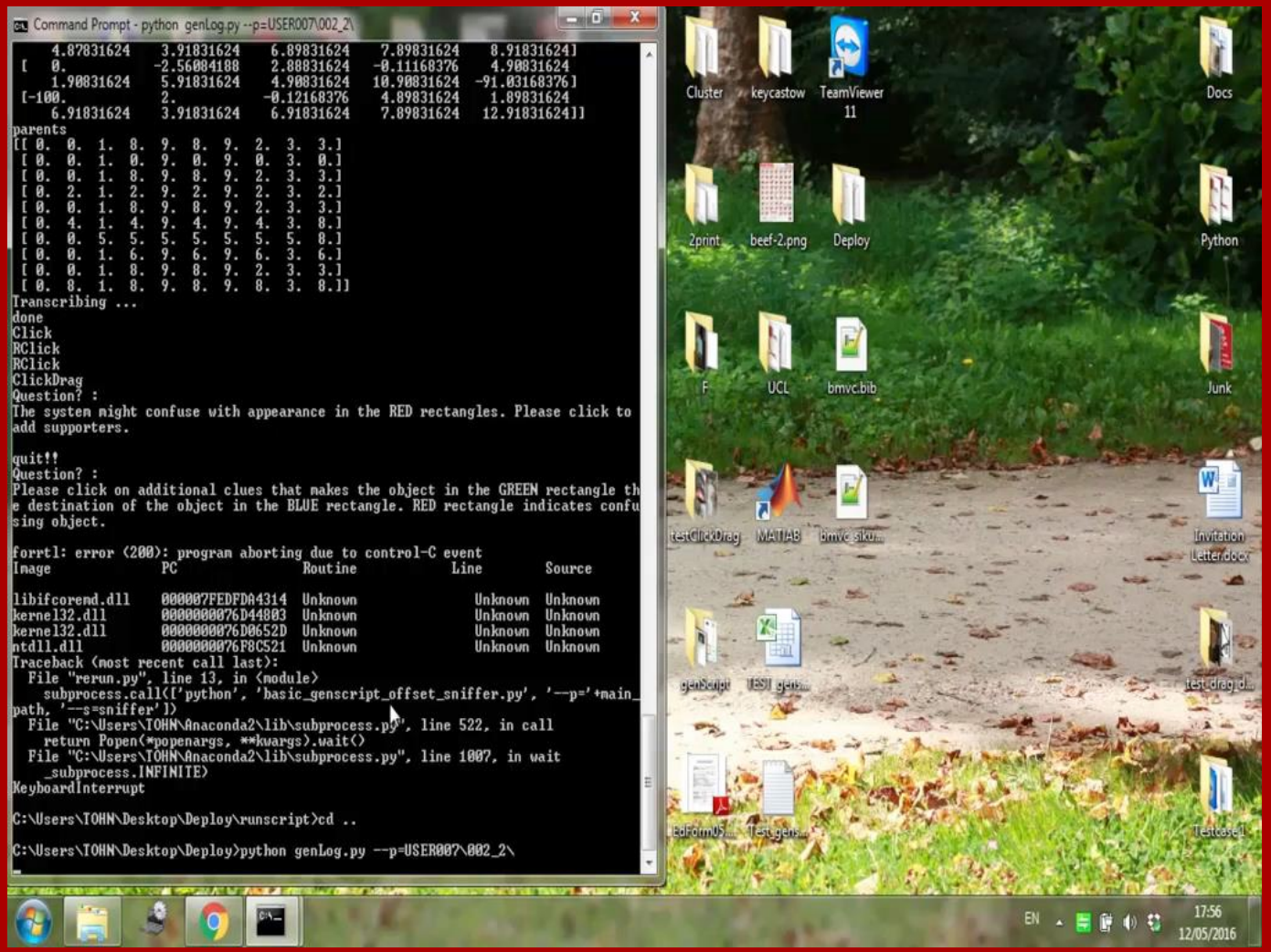
Phase A. HILC **observes** a user[1] **demonstrating** a task.

Phase B. HILC **asks a user**[2] to clarify confusing steps, if any, and then **synthesizes a task script**.

Phase C. A user[3] **triggers** the script to complete the task.

From User's Perspective

Phase A. HILC **observes** a user[1] **demonstrating** a task.



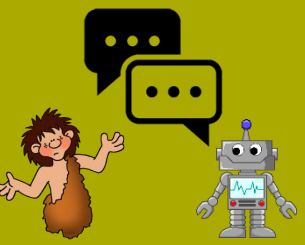
Instructor

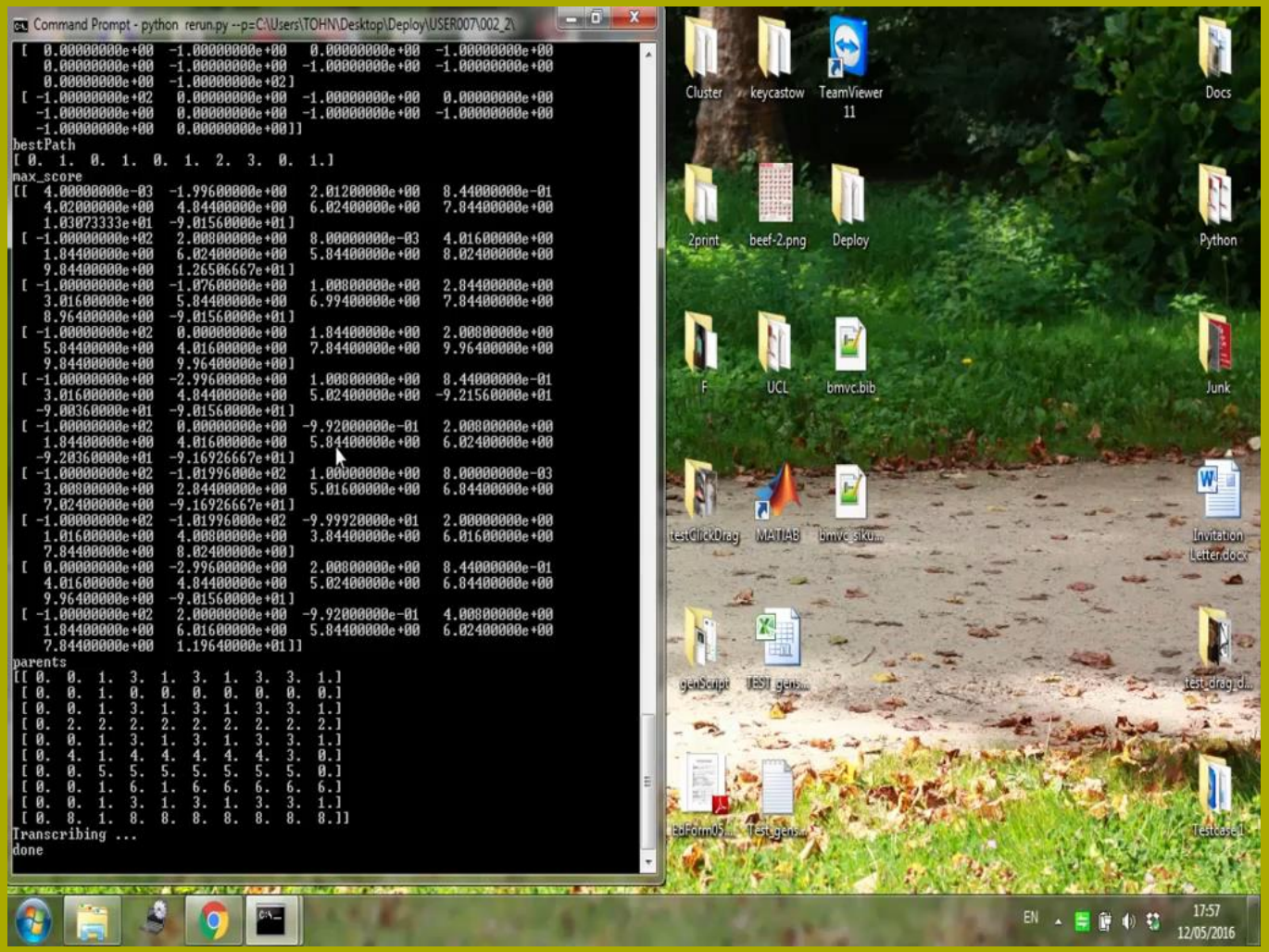
Teacher

End-user

From User's Perspective

Phase B. HILC **asks a user**[2] to clarify confusing steps.





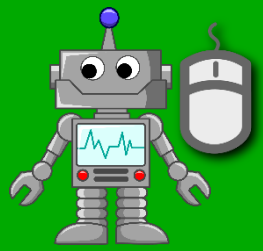
Instructor

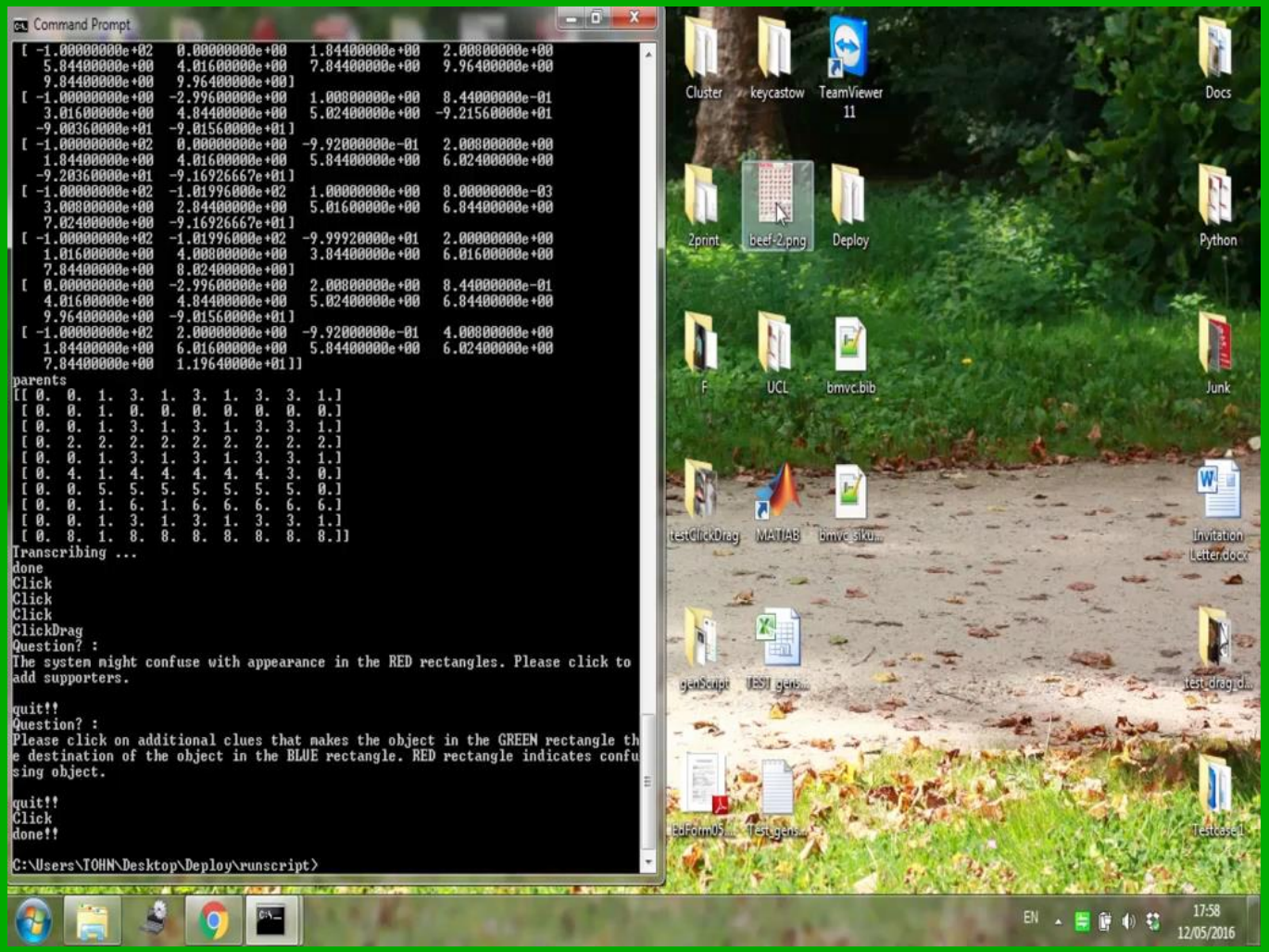
Teacher

End-user

From User's Perspective

Phase C. A user[3] **triggers** the script to complete the task.





Instructor

Teacher

End-user

From development perspective

Phases	Task Classification: Linear	Looping	Monitoring
A: Demonstration	Normal	+Signal	+Signal
B: Teaching	Supporters	Positives False positives Supporters	Visual Cues
C: Running	Execute	Execute	Execute

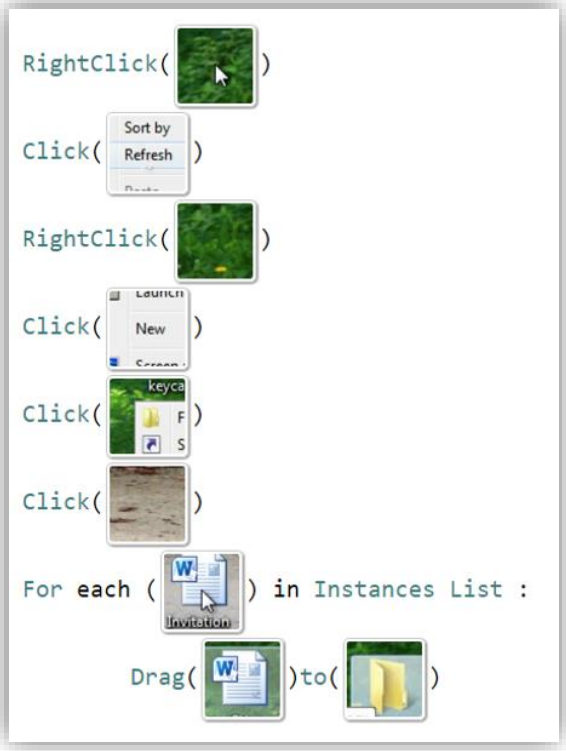
Phase A : Demonstration Phase

- Sniffer
- Video

Log-file

```
INFO:root:6195671,000000.bmp,1405,357,no press,
INFO:root:6195703,000001.bmp,1439,197,no press,
INFO:root:6195734,000002.bmp,1463,168,no press,
INFO:root:6195781,000003.bmp,1480,144,no press,
INFO:root:6195796,000004.bmp,1482,134,no press,
INFO:root:6195859,000005.bmp,1482,132,no press,
INFO:root:6195906,000006.bmp,1482,119,no press,
INFO:root:6195937,000007.bmp,1482,117,no press,
INFO:root:6196046,000008.bmp,1483,123,no press,
INFO:root:6196062,000009.bmp,1503,168,no press,
INFO:root:6196093,000010.bmp,1569,226,no press,
INFO:root:6196140,000011.bmp,1593,241,no press,
INFO:root:6196171,000012.bmp,1595,242,no press,
INFO:root:6196296,000013.bmp,1595,243,no press,
INFO:root:6196343,000014.bmp,1606,292,no press,
INFO:root:6196375,000015.bmp,1620,316,no press,
INFO:root:6196484,000016.bmp,1620,317,no press,
INFO:root:6196531,000017.bmp,1620,320,no press,
INFO:root:6196562,000018.bmp,1619,321,no press,
INFO:root:6196671,000019.bmp,1618,321,no press,
INFO:root:6196718,000020.bmp,1608,317,no press,
INFO:root:6196750,000021.bmp,1606,315,no press,
INFO:root:6196843,000022.bmp,1606,315,no press,
INFO:root:6196875,000023.bmp,1603,313,no press,
INFO:root:6196921,000024.bmp,1602,312,no press,
INFO:root:6196984,000025.bmp,1601,310,no press,
INFO:root:6197125,000026.bmp,1601,310,press left,
INFO:root:6197250,000027.bmp,1601,310,no press,
INFO:root:6197578,000028.bmp,1596,310,no press,
```

Sequence of basic actions



Phase B : Teaching phase

HILC gets follow-up input from user (a.k.a. Teacher):

- more positive examples
- remove false positive examples
- point out the visual cues
- add supporters

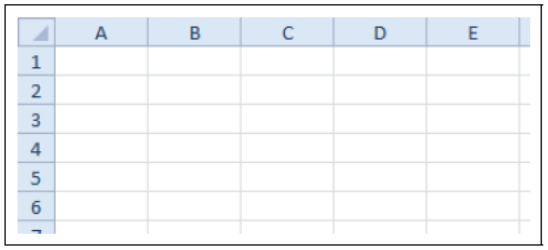
Phase B : Teaching phase

HILC allows users to add more positive examples, remove false positive examples, point out the visual and add supporters

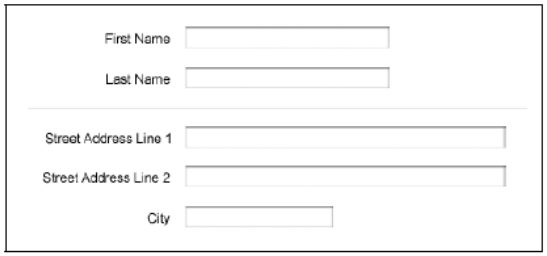
Supporters

(a)

(b)



(c)

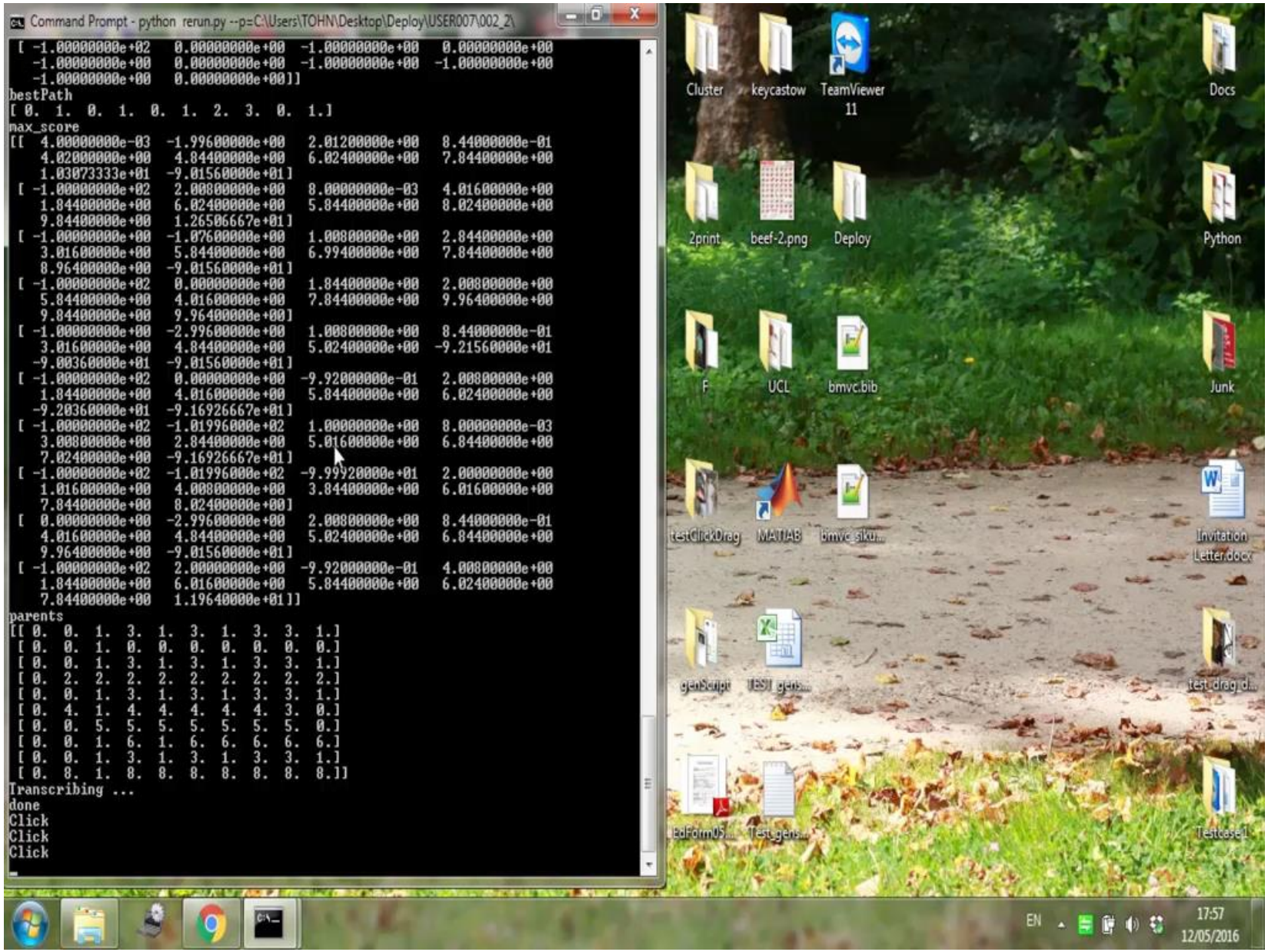


(d)

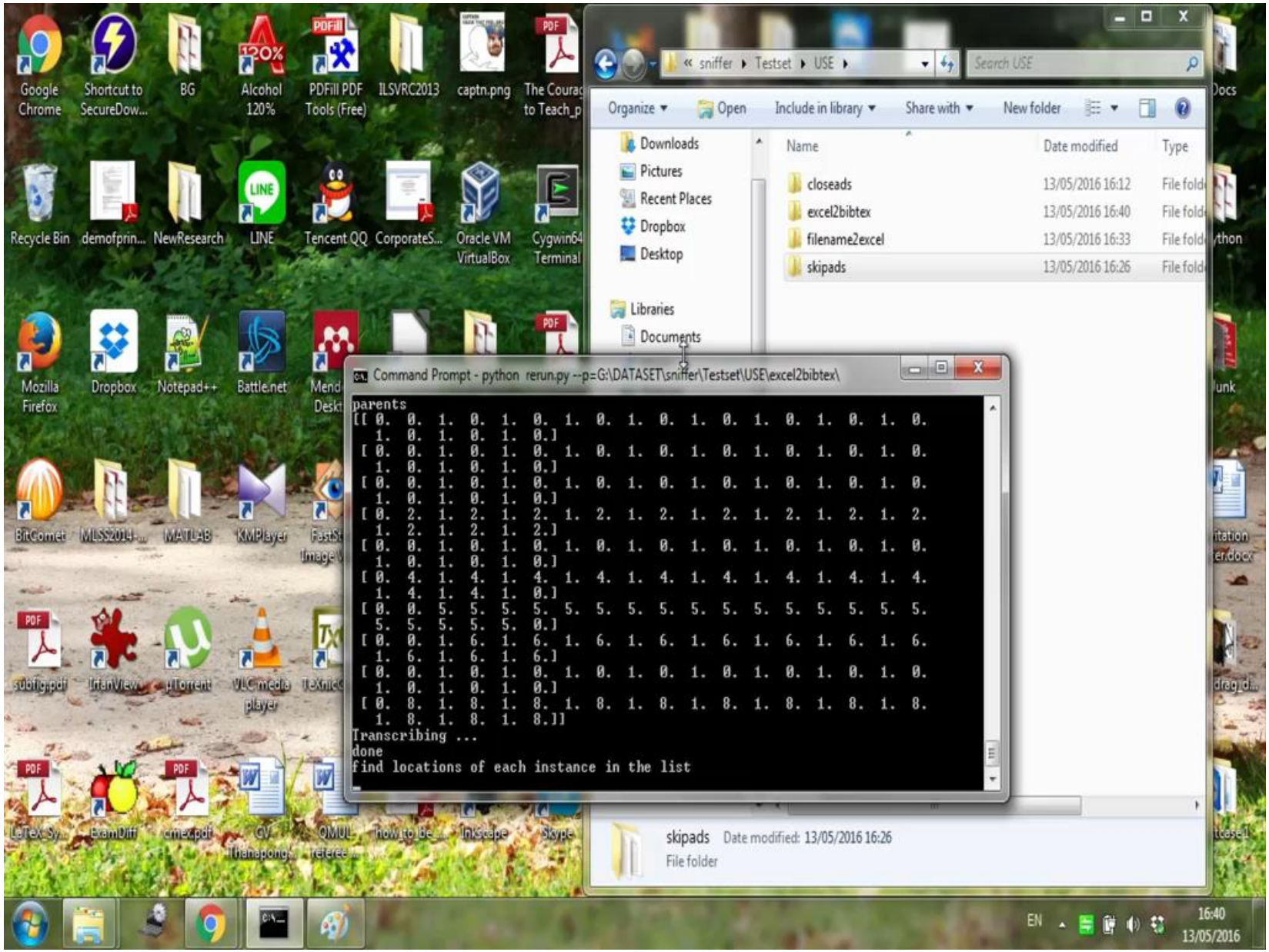
Teaching phase - How did we do it?

When and What to ask?

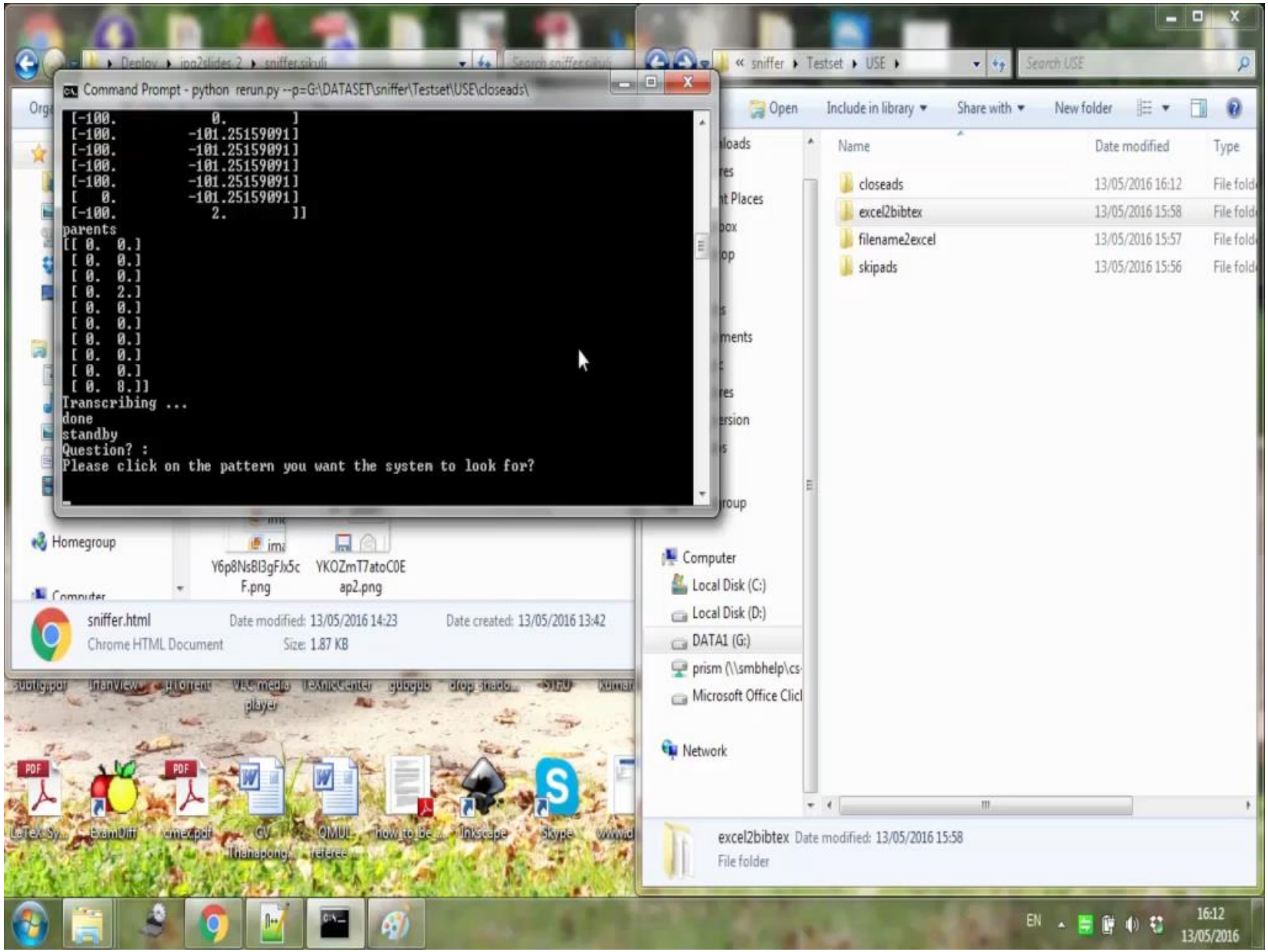
When and What to ask? (Linear tasks)



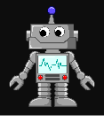
When and What to ask? (Looping task)

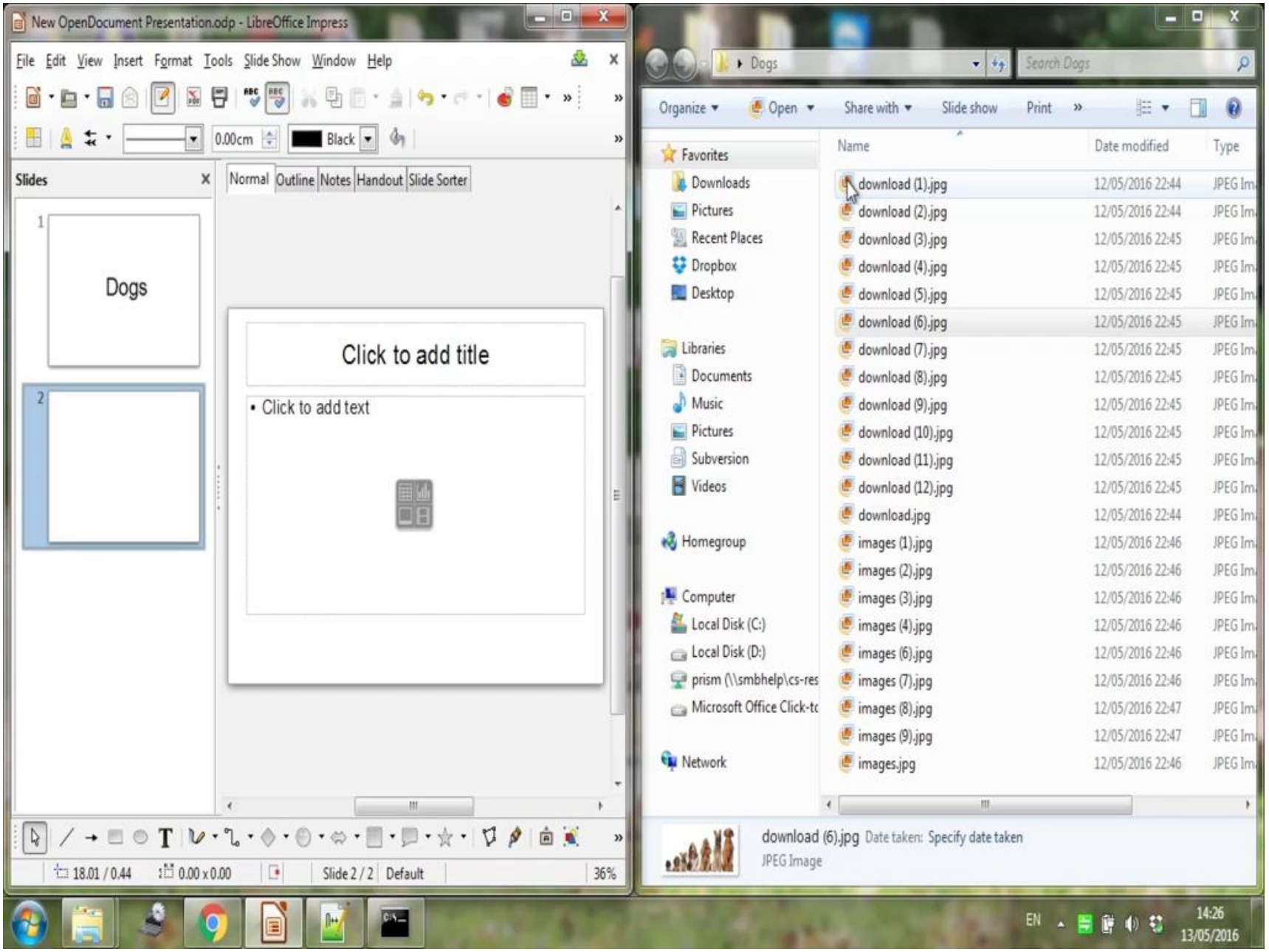


When and What to ask? (Monitoring task)



Phase C : Running phase





Scenarios and experiments

Scenario	Basic Actions + Typing					Transcription		Reproduction		Demonstration Time VS Refining Time (average)	
	Click	Click Drag	Double Click	Right Click	Typing	Sikuli Slides	Our	Sikuli Slides	Our	Sikuli Slides	Our
1 Mute Audio playback (Linear)	2	0	0	0	0	✓	✓	✓*	✓	10s/49s	10s/27s
2 Turn on High-Contrast-Mode (Linear)	6	1	0	0	0	✓*	✓	✓**	✓	27s/10m	27s/170s
3 Remote access with Team Viewer (Linear)	11	0	0	0	4	✓*	✓	✗	✓	40s/∞	40s/4m
3.2 Remote access with Team Viewer (Linear)	13	0	0	4	0	✓*	✓	✗	✓	37s/∞	37s/7m
4. Skip YouTube ads (Monitoring)	1	0	0	0	0	✗	✓	✗	✓	N/A	10s/5.5m
5. Close YouTube ads (Monitoring)	1	0	0	0	0	✗	✓	✗	✓	N/A	12s/6.9m
6. Create slides out of jpgs folder (Looping)	2x	1x	0	0	0	✗	✓	✗	✓	N/A	35s/10m
7. Create spreadsheet of filenames (Looping)	4x	2x	0	0	4x	✗	✓	✗	✓	N/A	60s/6.6m
8. Create BibTex from spreadsheet (Looping)	9x	0	0	0	8x	✗	✓	✗	✓	N/A	86s/12.5m
9. Move MSWord files to a folder (Looping-Video)	4	1x	0	2	0	✗	✓	✗	✓	N/A	25s/**22m**

Table 1. User study on our system compared to Sikuli Slides. Scenario 3.2 is an alternative way to perform Scenario 3, without pressing shortcut key combinations that Sikuli Slides is known to be missing. Nevertheless, we eventually realized that Sikuli Slides isn't detecting the right click actions either. (✓= successful, ✓* = partially successful, ✓ = can be successful with guidance from the operator, ✗= can not succeed at the task at all). x represents the number of repeated loops needed to complete the task. Please note that 90% of the refining time for Task 9 is offline - devoted to the time spent on processing video to produce the log-file.**

Limitations

- HILC is not aware of off-screen objects.
  because it only looks at a screenshot image.
- Scrolling is currently not supported.
- HILC is not aware of system state, hence it only relies on a fixed time delay between basic actions.
- HILC relies on having a sniffer, but we have preliminary results from just videos of the GUI.

Take home messages

- Computer Vision.
- 3 Phases.
- Follow-up questions.

Thank you

Q & A

For more information and code please visit http://visual.cs.ucl.ac.uk/pubs/HILC/

Extra slides

Document Object Model

From Wikipedia, the free encyclopedia

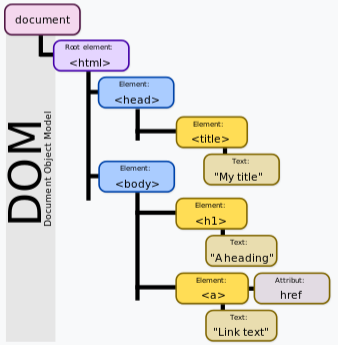
This article includes a list of references, but **its sources remain unclear** because it has **insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(August 2010)* *(Learn how and when to remove this template message)*

The **Document Object Model** (**DOM**) is a cross-platform and language-independent application programming interface that treats an HTML, XHTML, or XML document as a tree structure wherein each node is an object representing a part of the document. The objects can be manipulated programmatically and any visible changes occurring as a result may then be reflected in the display of the document.[2]

Principal standardization of DOM was handled by the W3C, which last developed a recommendation in 2004. WHATWG took over development of the standard, publishing it as a living document. The W3C now publishes stable snapshots of the WHATWG standard.

Contents [hide]

Document Object Model



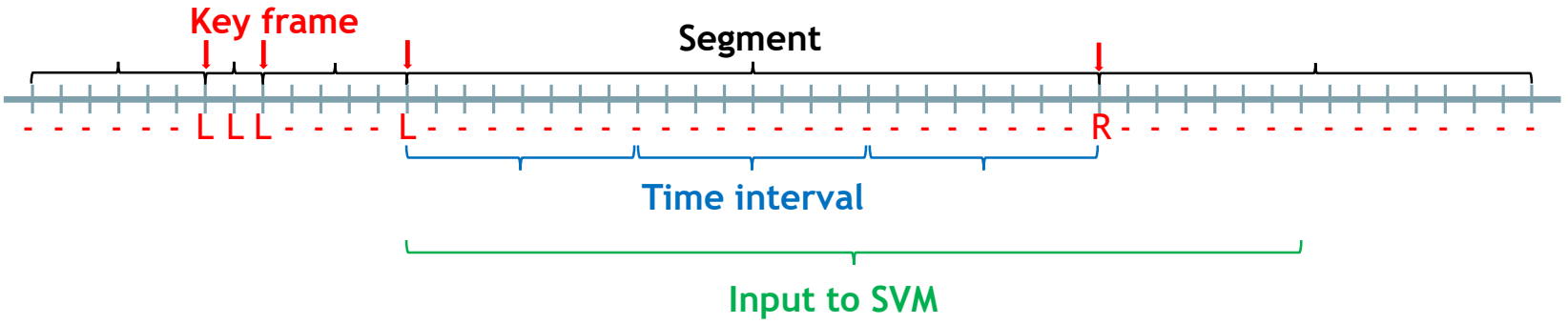
Example of DOM hierarchy in an HTML document

First published	October 1, 1998; 18 years ago
Latest version	DOM4[1]

Demonstration Phase - How did we do it?

- The log file unifies inputs from both video and sniffer

- Transcribe input log-file
  - Viterbi Algorithm
    - Unary potential is computed using SVM + Random Forest
    - Pairwise potential encourages segments from the same basic action to follow the order



Basic actions = {Click, Double Click, Click Drag, Right Click}