Lecture 8:
Designing complex systems

Case studies on applying theory to hard HCI problems

Overview of the course

- Theory driven approaches to HCI
- Design of visual displays
- Goal-oriented interaction
- Designing smart systems (guest lecturer)
- Designing efficient systems
- Designing meaningful systems (guest lecturer)
- Evaluating interactive system designs
- **Designing complex systems**

What are some things that make designs complex?

- How complex is the domain?
- How many different tasks might a user perform?
- How well defined are the outcomes? (Wicked problems, L3)
- How easy is it to understand each part?
- When the parts are put together how easy is to guess the behaviour?
- Does the system do things when the user isn’t there? (Attention Investment from L3)

Designing tasks vs interaction spaces

Consider a (slightly silly) APIs for sending a message:

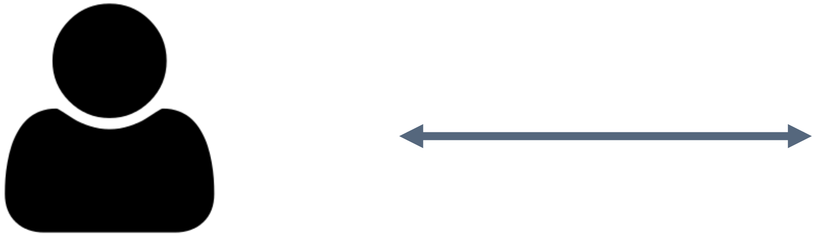
```
(1)sendTheRightMessage()
(2)sendMessage(Enum message)
(3)sendMessage(String message, Urgency status)
```

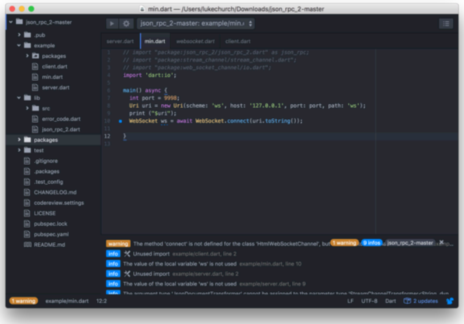
- Naive design would result (1). Complex systems tend to be built out of reusable components that the users configure (2,3)
- Building this kind of system involves discussing tradeoffs as well as detailed design decisions
- This is the kind of system that most of you will build: **Programming languages, APIs, AI systems**

Broad brush techniques

- Descriptions of specific actions result in a ‘death by detail’
- Don’t describe specific actions with an interface
  - Describe interaction with a level of *analytical distance* from the interface
  - Use an *analytical frame* which is a way of structuring a description of an interaction
  - The description can then be compared to an ideal for a domain to become a critical perspective (see Lecture 1)
- These techniques often give names to the patterns

Cognitive Dimensions of Notations (CDNs): Analytical Frame





```
// import "package:json_rpc_2/json_rpc_2.dart" as json_rpc;
// import "package:stream_channel/stream_channel.dart";
// import "package:web_socket_channel/io.dart";
import 'dart:io';

main() async {
  int port = 9998;
  Uri uri = new Uri(scheme: 'ws', host: '127.0.0.1', port: port, path: 'ws');
  print ("$uri");
  WebSocket ws = await WebSocket.connect(uri.toString());

}
```

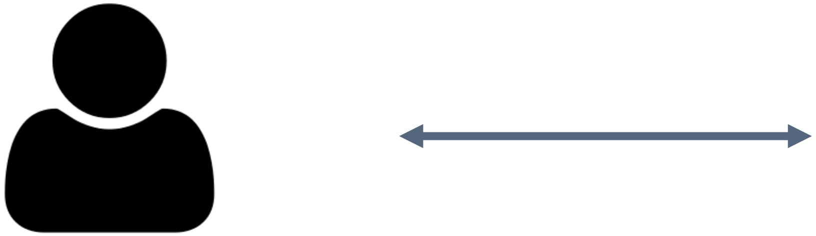
```
warning The method 'connect' is not defined for the class 'HtmlWebSocketChannel', but
info Unused import example/client.dart, line 2
info The value of the local variable 'ws' is not used example/min.dart, line 10
info Unused import example/server.dart, line 2
info The value of the local variable 'ws' is not used example/server.dart, line 9
```

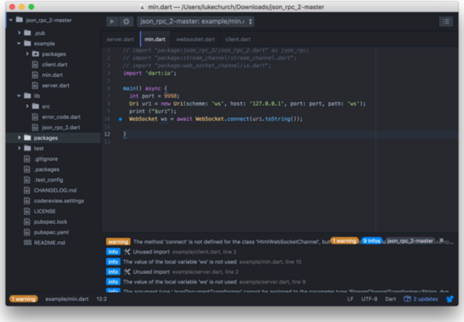
A user

Performs an *activity*

Interface containing *notations,* described along a number of *dimensions*

Cognitive Dimensions of Notations (CDNs): Analytical Frame





```
1  // import "package:json_rpc_2/json_rpc_2.dart" as json_rpc;
2  // import "package:stream_channel/stream_channel.dart";
3  // import "package:web_socket_channel/io.dart";
4  import 'dart:io';
5
6  main() async {
7    int port = 9998;
8    Uri uri = new Uri(scheme: 'ws', host: '127.0.0.1', port: port, path: 'ws');
9    print ("$uri");
10   WebSocket ws = await WebSocket.connect(uri.toString());
11
12 }
13
```

warning The method 'connect' is not defined for the class 'HtmlWebSocketChannel', but

info Unused import example/client.dart, line 2

info The value of the local variable 'ws' is not used example/min.dart, line 10

info Unused import example/server.dart, line 2

info The value of the local variable 'ws' is not used example/server.dart, line 9

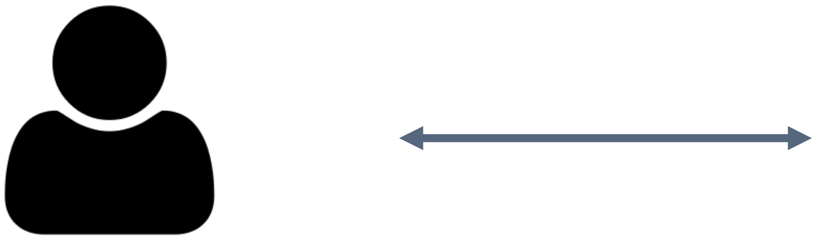
info The argument type 'JsonDocumentTransformer' cannot be assigned to the parameter type 'StreamChannelTransformer<String, dyn

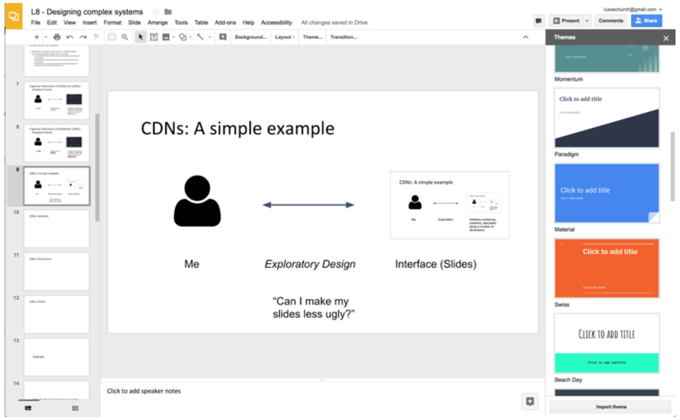
A user

Performs an *activity*

Interface containing *notations,* described along a number of *dimensions*

CDNs: A simple example





CDNs: A simple example

Me

Exploratory Design

Interface (Slides)

"Can I make my slides less ugly?"

Me

Exploratory Design

Interface (Google Slides)

"Can I make my slides less ugly?"

CDNs: A simple example (Demo)

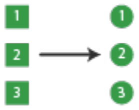
- One described change "Make the font of the headings Comic Sans"
  - Select the first slide, change the font
  - Select the second slide, change the font
  - Yawn.
- This is repetition *Viscosity*, many operations to perform one change
- Design maneuver: Introduce an *Abstraction* (master slide), decreases *Viscosity*, but increases *Premature Commitment*
- NB: CDNs analysis is meaningless independent of an interface.

CDNs: Activities

EXPLORATION

Manipulating both information and structure

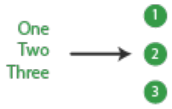
Exploration involves manipulating, and changing, both the content and the structure of the information



MODIFICATION

Changing structure only

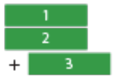
Modification is changing the structure of the information, but keeping the content the same. Also known as 'refactoring'



TRANSCRIPTION

From one notation to another

Transcription involves copying information from one notational form to another, often between different media as well



INCREMENTATION

Adding one more

Incrementation is adding new content, but leaving the information structure alone

READING

Seeking information or gist

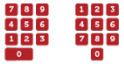
Reading doesn't involve changing the content or structure, but finding information either for detail or an overview

CDNs: Dimensions

ROLE EXPRESSIVENESS

How much elements suggest their purpose

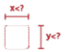
Some syntax suggests its purpose better than others, this can help learning a system but is difficult to achieve without accepting the limitations of existing conventions



CONSISTENCY

Similar meanings, similar syntax

Internal consistancy is important for understanding systems



PREMATURE COMMITMENT

Constraints on the order of decisions

Constraints on order often involve people having to make decisions before they would normally do in the course of solving a problem. This pre-thinking obstructs exploration, but can cause Useful Awkwardness

VISCOSITY

Resistance to change

One change in the mind becomes many operations in the interface. Viscosity is commonly exchanged for an abstraction.

ADD AGE TO NEW_AGE GIVING NEW_AGE

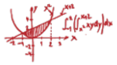
vs

new_age += age

DIFFUSENESS

The spread-out-ness of information

Information that is very diffuse can make it easier to see but constrains the amount that can be seen at any one time, decreases juxtaposition and can add to viscosity



USEFUL AWKARDNESS

Thinking hard is sometimes useful

Some notations cause the user to have to deeply consider their domain, which sometimes results in useful results. This is often caused by Premature Commitment.

ABSTRACTION

Mechanisms for generality

Abstractions support operations over multiple objects, or when the user isn't present Abstractions provide support for efficient use, but may increase cognitive load, user perception of risk, and premature commitment

SYNOPSIE

Provides an understanding of the whole

Some notations provide a sense that you can step back and get a holistic impression, also described as the 'gestalt view'

HIDDEN DEPENDENCIES

Unexpected relationships

When one item is changed another, seemingly unconnected, item changes. Harmful to exploration. Commonly reduced by making the dependencies visible, at the expense of diffuseness and viscosity

HARD MENTAL OPERATIONS

Some things are just Hard

Some tasks that are known to be cognitively challenging, for example remembering lists of different possible branches

PROVISIONALITY

Degree of commitment to marks

High provisionality supports exploratory strategies such as playing 'what-if' games but can increase viscosity by expecting users to state when they are ready to 'commit' to their marks

(<[^>]*>)/1

LEGIBILITY

Readability of the notation

Various factors affect a notations readability such as how distinguishable its characters are and how well it supports perceptual parsing. Often trades off with diffuseness

CLOSENESS OF MAPPING

Correspondence to the domain being expressed

A close relationship between the notation and the domain that it models makes implementation easier but may result in inflexibility and diffuseness

PROGRESSIVE EVALUATION

Feedback along the way

Progressive Evaluation describes how much the system displays partial progress towards a goal, or mandates the whole thing to be finished before any feedback is given.

SECONDARY NOTATION

Escape from formality

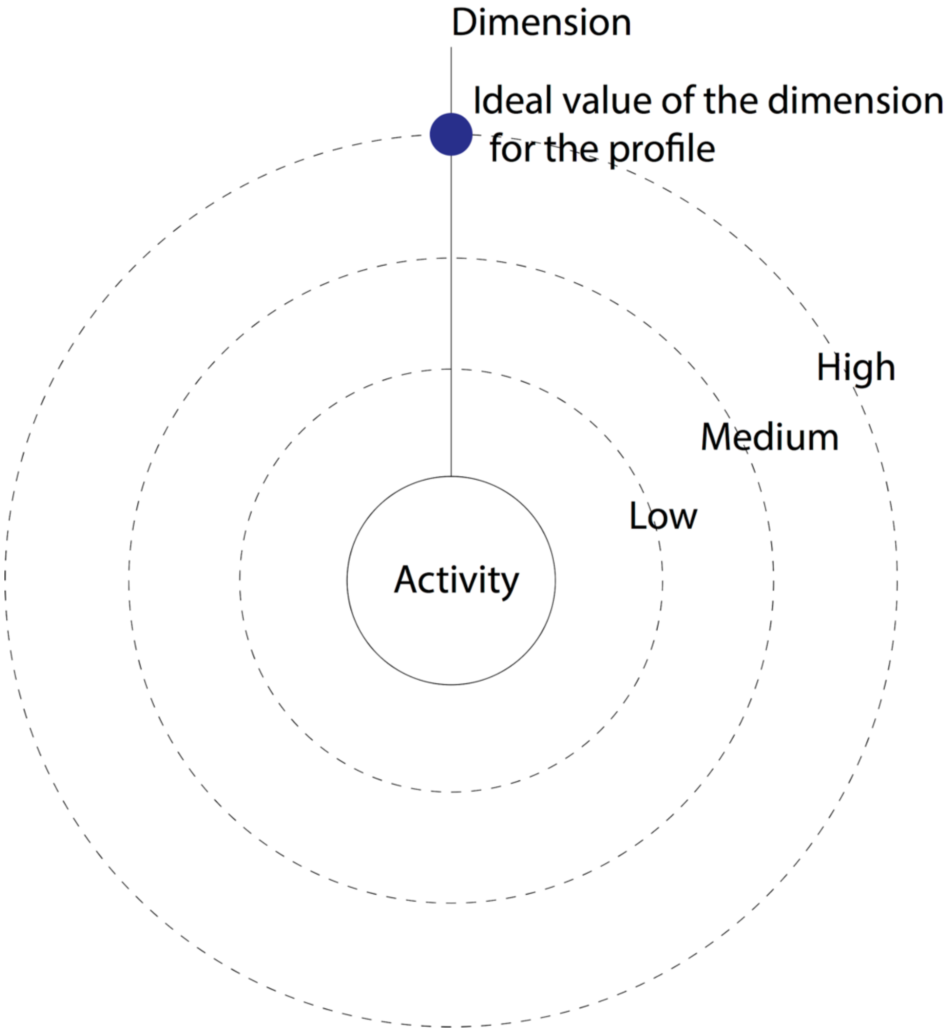
Notation that is not formally interpreted. Comments, notes, layout etc. Often omitted from computer based systems

JUXTAPOSITION

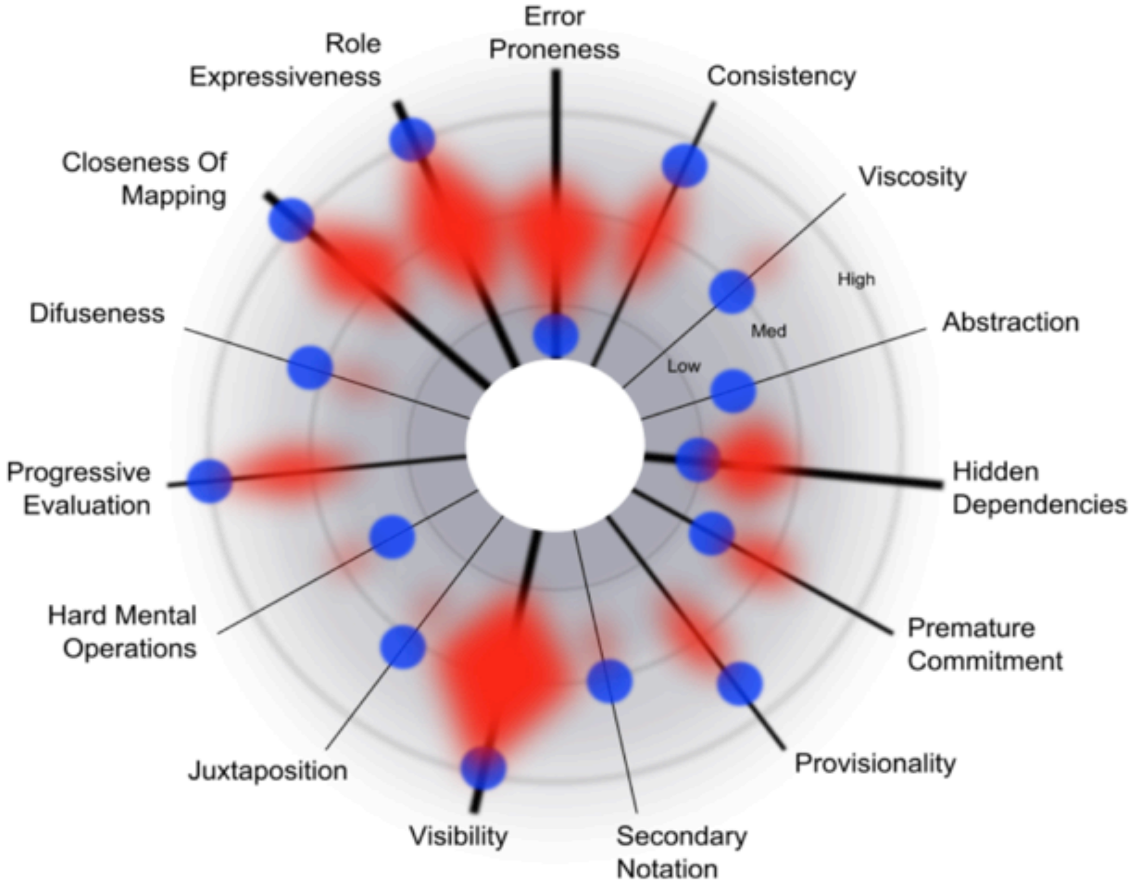
Simultaneous comparison

Simultaneous views of an information structure

CDNs: Profile



CDNs: Profile



Case Study: Dynamo’s type system

Dynamo

- Language for exploring building designs
- Live Demo
- Includes a constructor **Point(x, y, z)** and array literal syntax [1,2]

Design question for discussion:

"What should **Point([0, 1, 2], 10, 10)** do?"

- What activities are important?
- How important: Viscosity? Premature commitment? Hidden Dependencies? Abstraction hunger?

Case Study: Interaction with Machine Learning

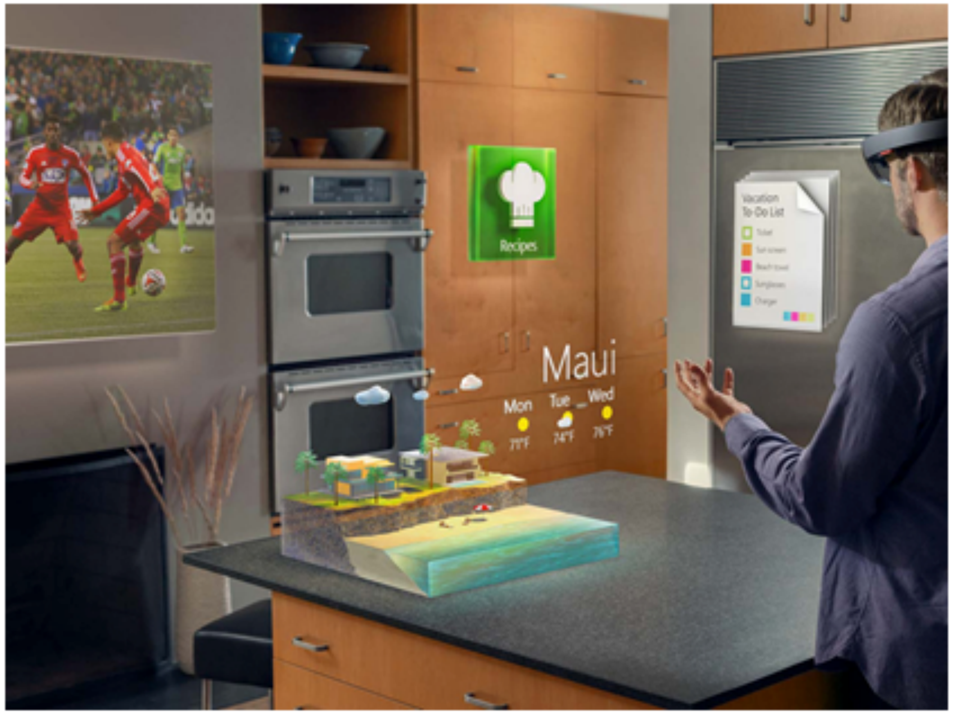
FlashFill for Excel

- From 2011 research by Sumit Gulwani at Microsoft Research
- “Synthesises a program from input-output examples”
    - How do you choose the examples?
    - How do you know what will happen?

- Live Demo (requires Excel 2013/16)
    - Paste a list of semi-structured text data into the left column
    - Type an example transform result in top cell to the right, then <Enter>
    - Press <Ctrl+E>

Some current research problems

Augmented reality is still a visual representation (remember metaphor?)



Microsoft Hololens

Programming, or direct manipulation?

- Many Internet of Things (IoT) devices have physical switches etc
  - But how do you define configuration, policy, future action?
  - Now we need a notation - or a programming language
- Remember behavioural economics and attention investment
  - Even around your house, bounded rationality happens

Philips Hue Light control

Conversational agents

- Do they build a user model, goal model or task model?
- Will this be more or less complex than FlashFill?
- How can you see it the model?
    - i.e. what is the notation?
- How could you modify the model?
    - … in response to errors (yours, or the system's)
    - … if you change your goals?
- Does having a 'body' help?
    - (remember metaphor)

Amazon Echo / Alexa agent

Human issues in machine learning

- Ethics and accountability
  - automating and/or justifying bias and prejudice
- Digital humanities
  - treating text and images as meaningful and sophisticated
  - (rather than just statistical fodder)
- Reward
  - who does the intellectual ‘work’ of providing training corpus content, data labelling, how are they paid, and where do the profits go?

Global challenges

- Is knowledge infrastructure built to ...
  - … prioritise low income populations
  - … advance United Nations Sustainable Development Goals (human rights, education etc)?

Africa's Voices Foundation / Cambridge Global Challenges Initiative

Further interest...

- Part II: Project
- Part III: Interacting with Machine Learning
- Research Skills: Working with artists and designers