Interaction Techniques 2

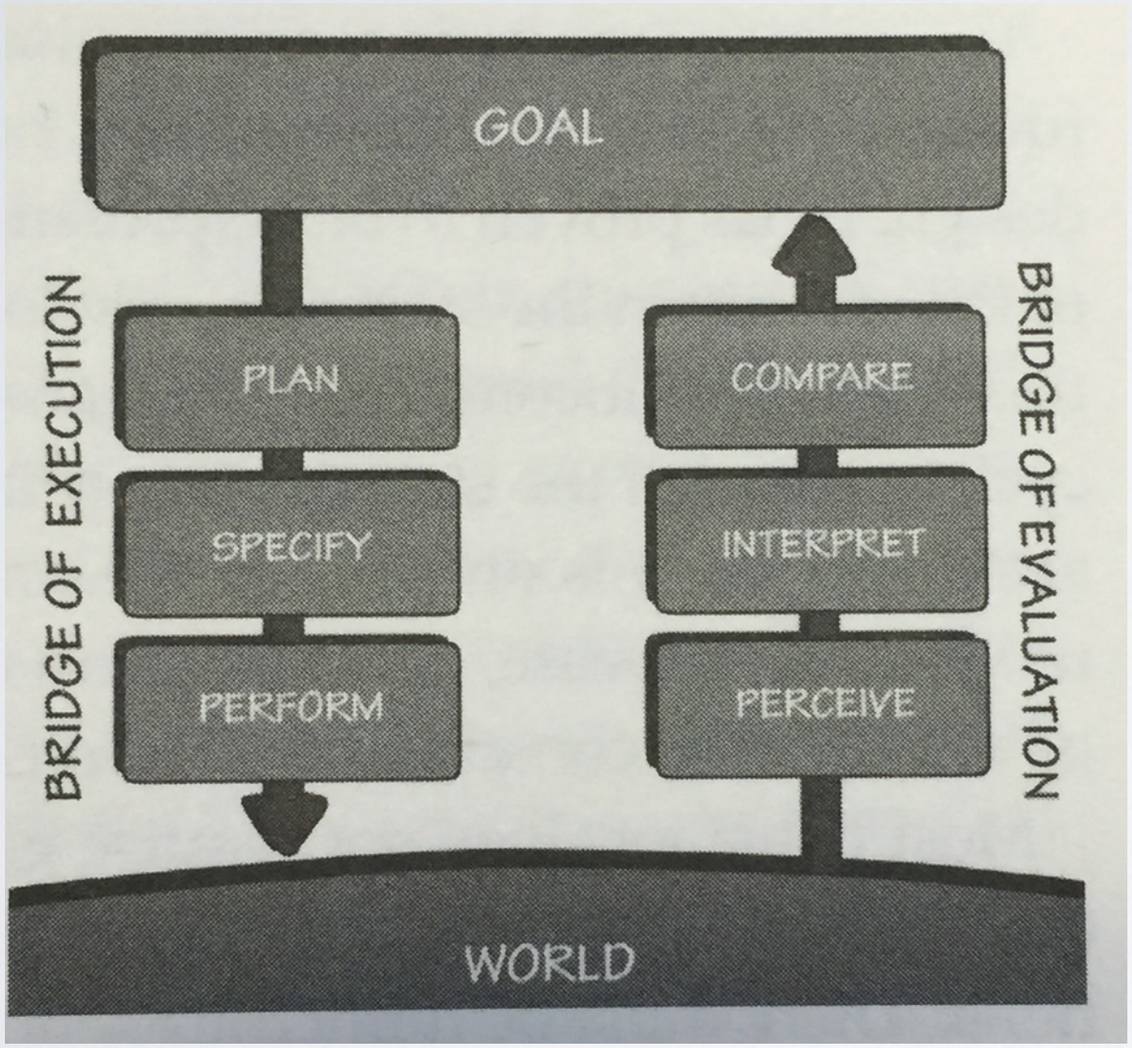
SWE 632
Fall 2015

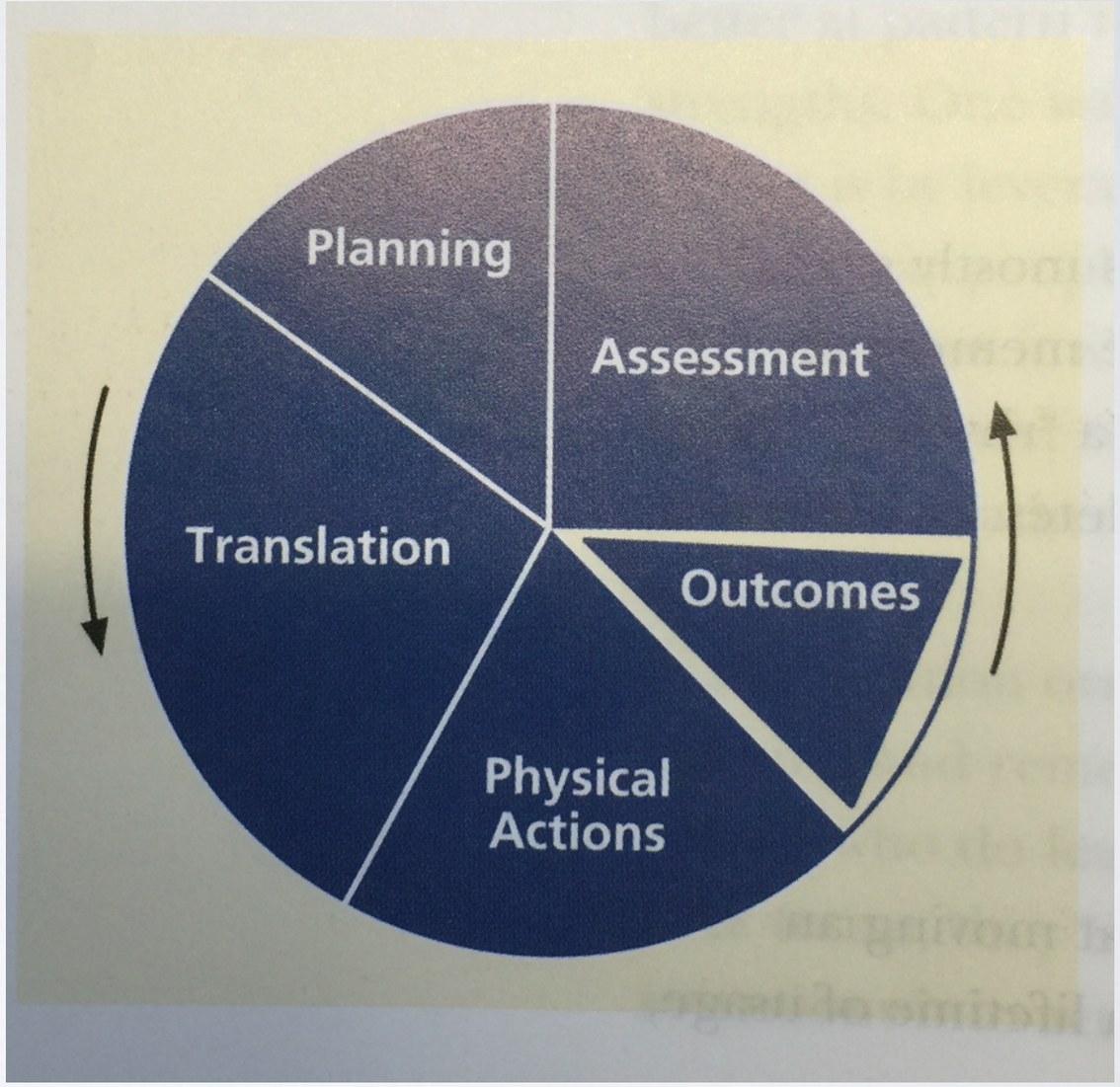
Administrivia

- HW 6 due today
- No class next week (Thanksgiving)
- HW 7 due on 12/3

Task structure

Hartson & Pyla Interaction Cycle



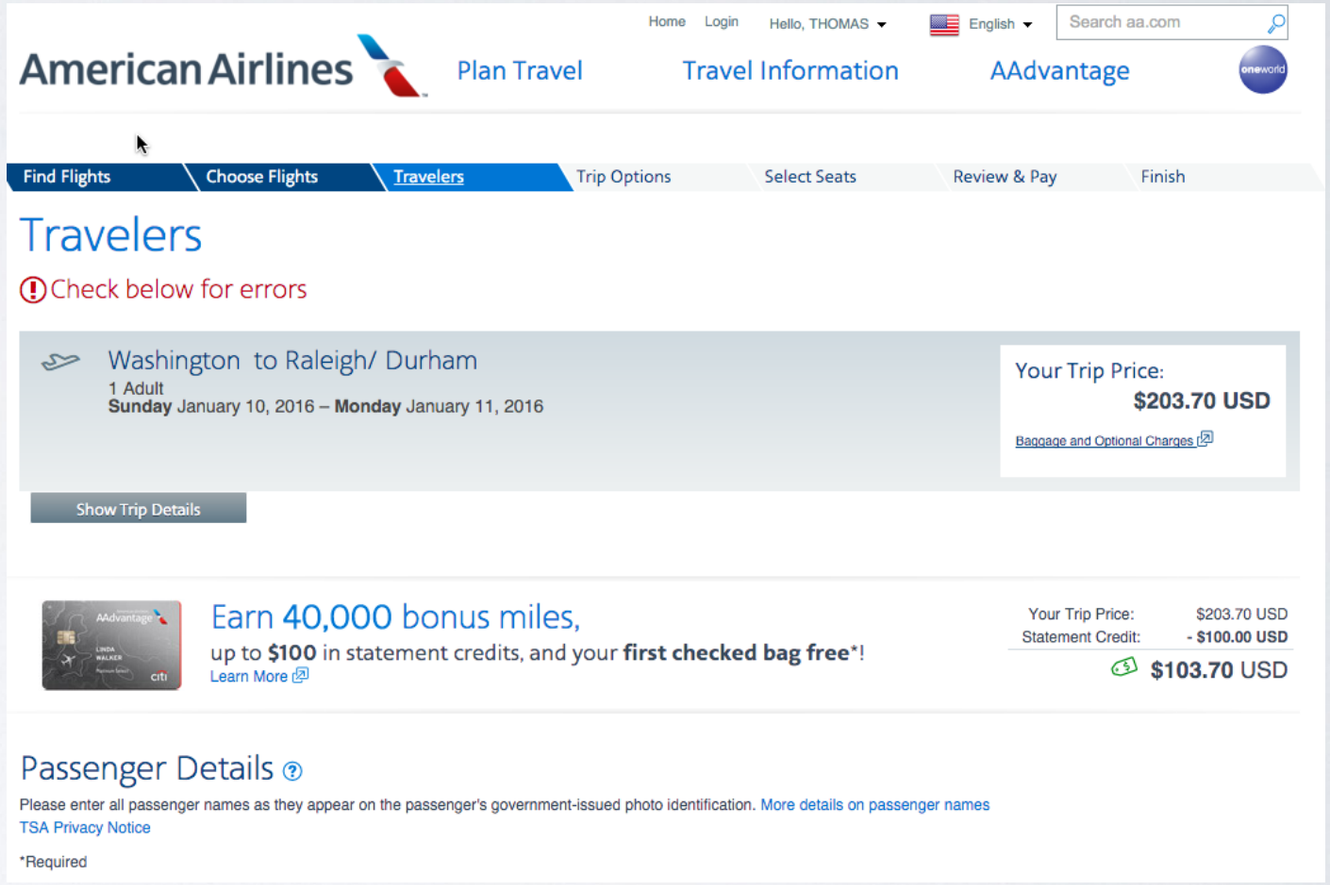


Task structure

- Flow of tasks and task steps
- Task design simplicity, flexibility, efficiency
- Maintenance of locus of control
- Direct manipulation

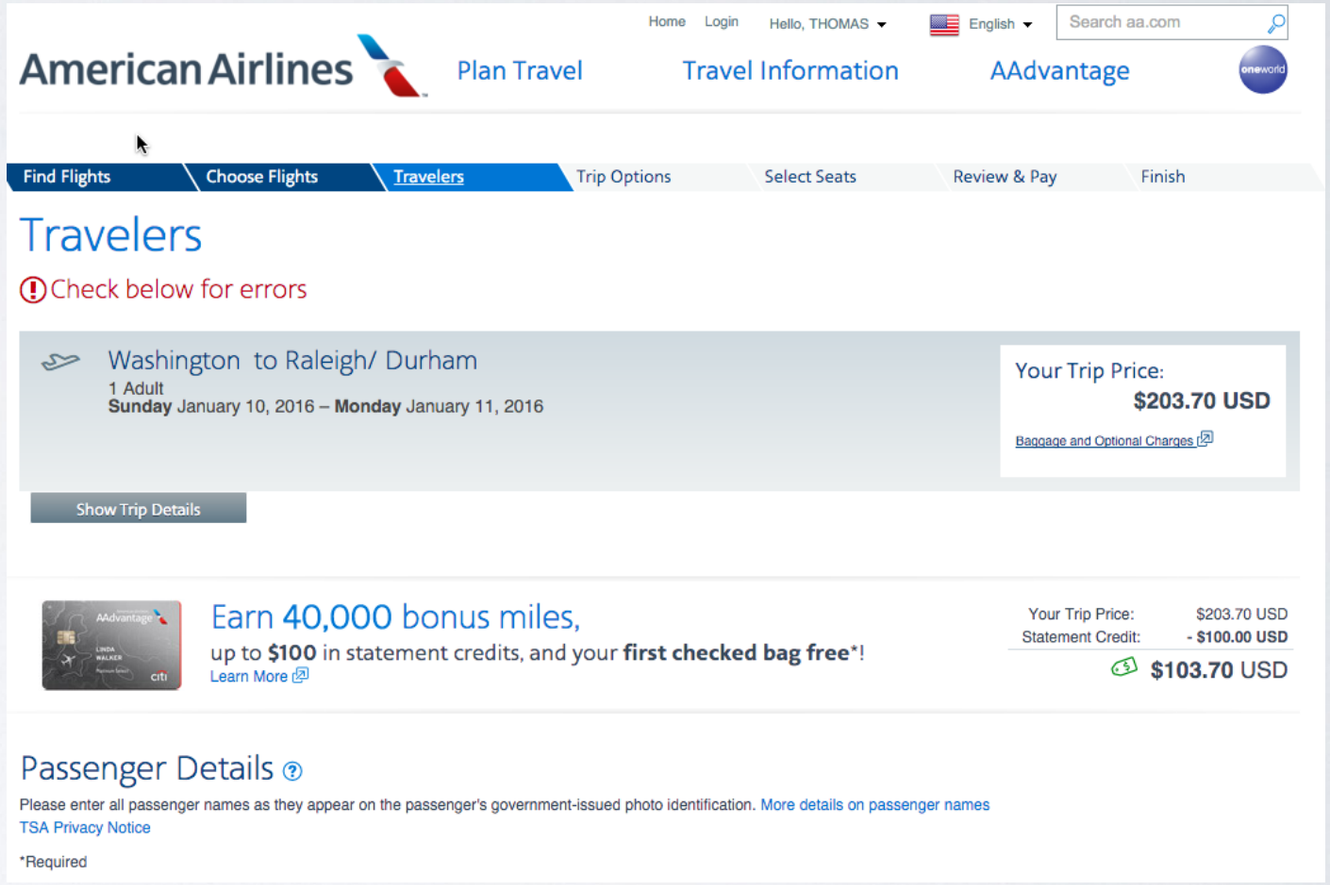
Separate long tasks into sequences

- Reduce STM demands by having user only work on one aspect of larger task at a time
- Don't interrupt users in the middle with unrelated tasks
- Provide closure of each subtask at the end



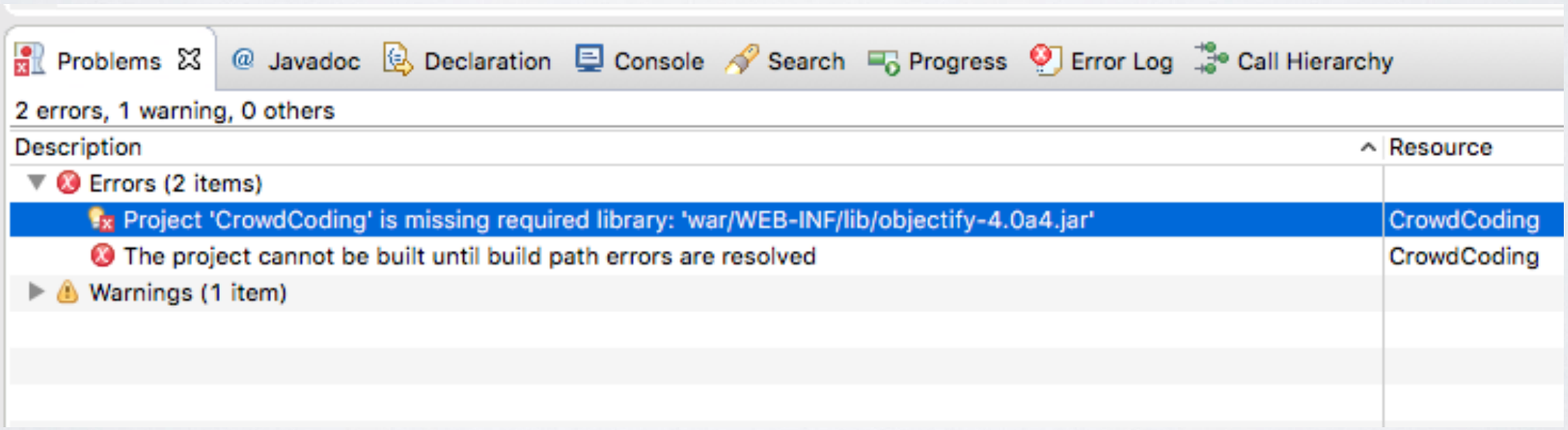
Design for flexibility & efficiency

- Users may take paths never envisioned by designer
- Using studies to identify different task flows, design flexible support for each



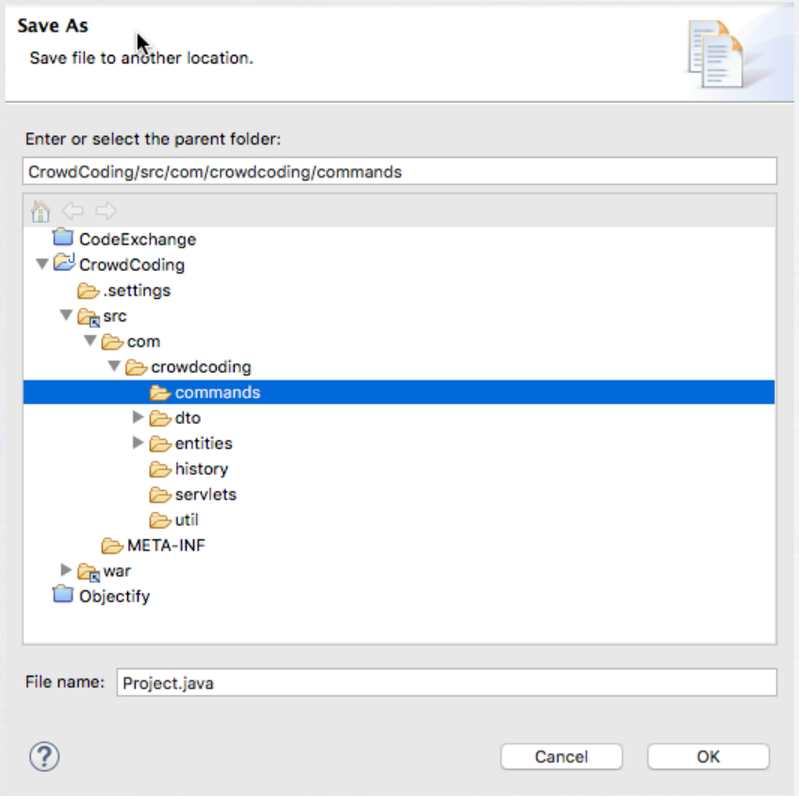
Show users how to fix errors

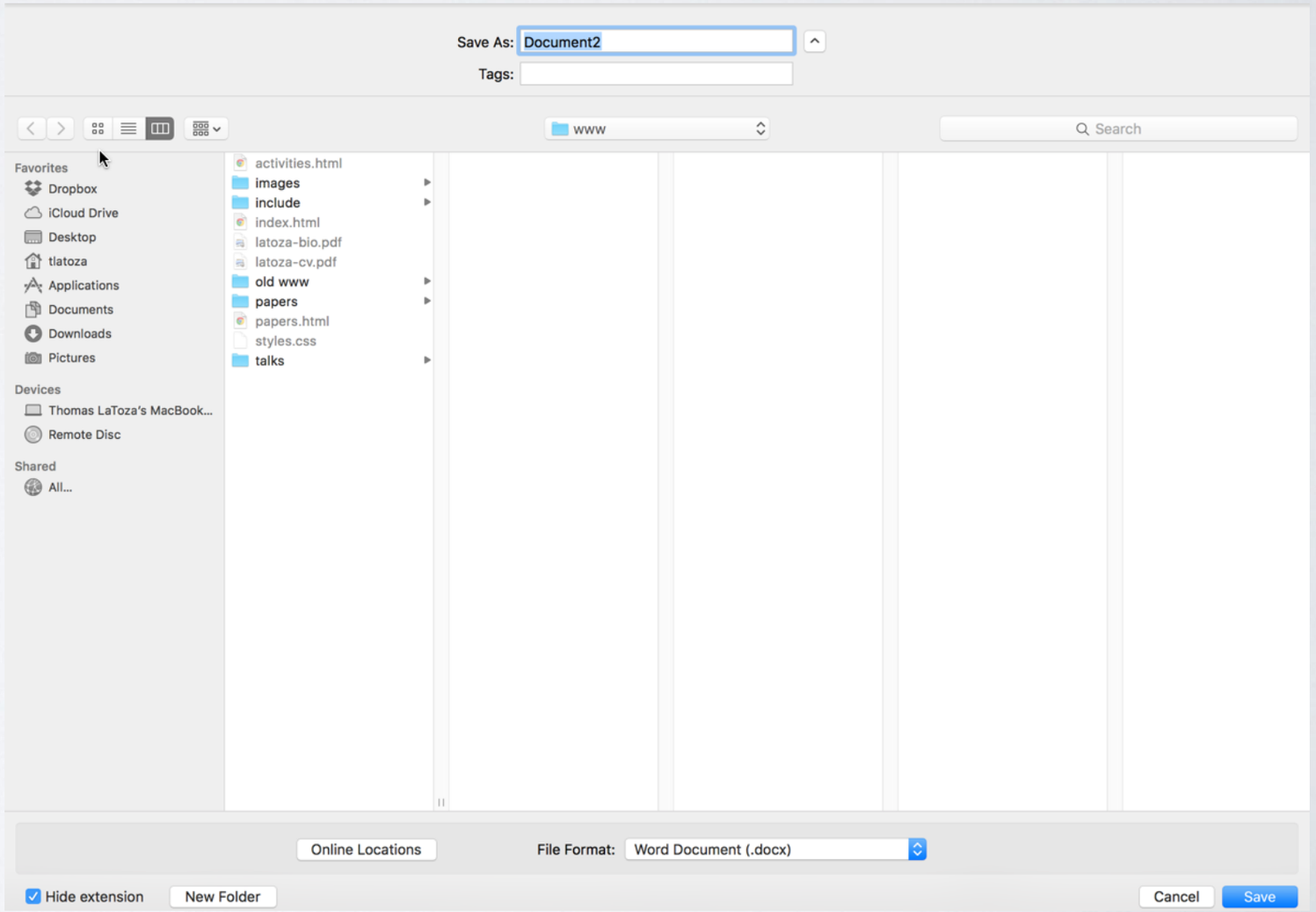
- Good: detecting user errors
- Better: directly showing how errors can be fixed
- (Best: using constraints to prevent errors from ever occurring)



Anticipate likely next actions

- Based on typical observed task flows, surface options for user to take likely next steps

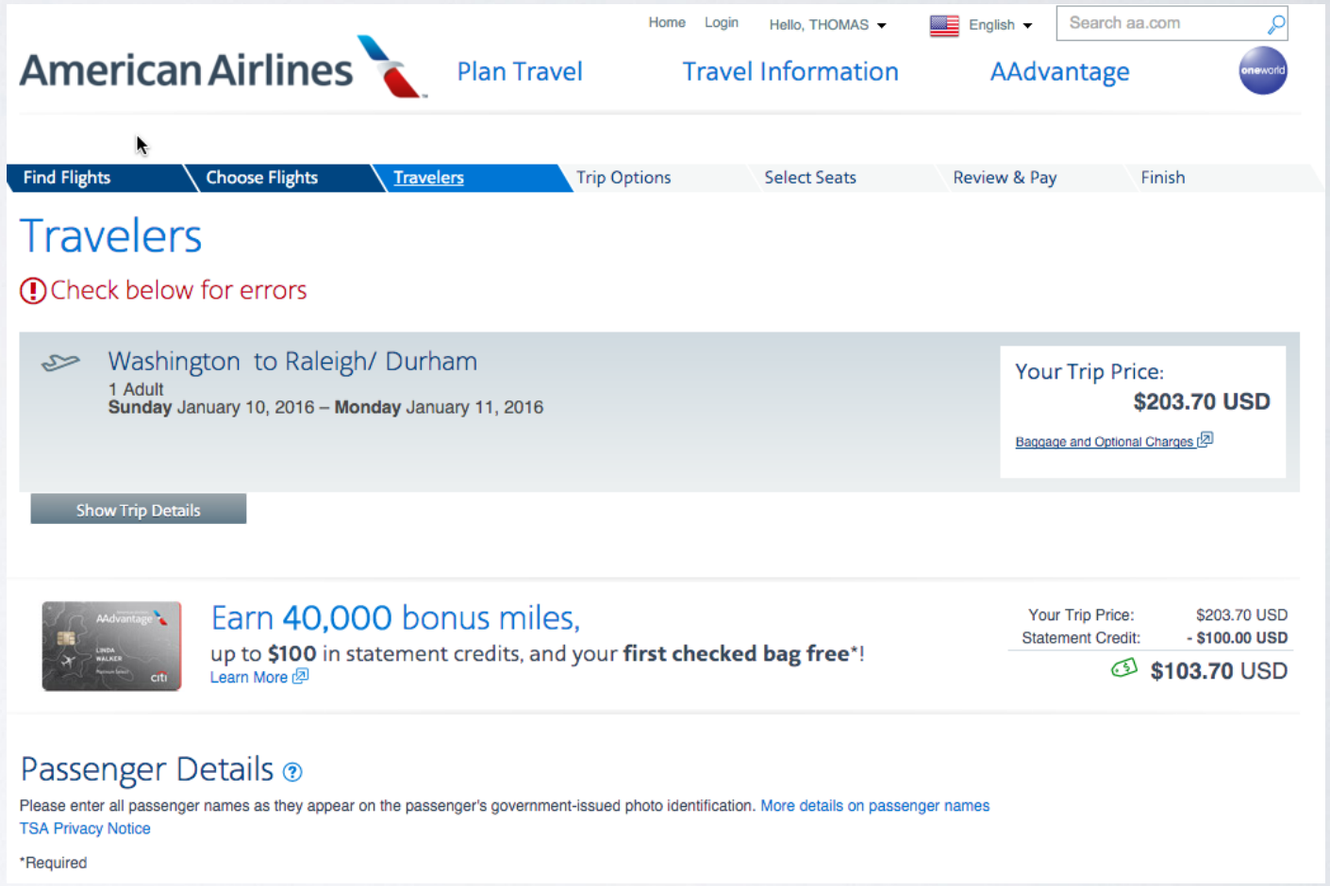




what if folder does not exist?

Keep users in control

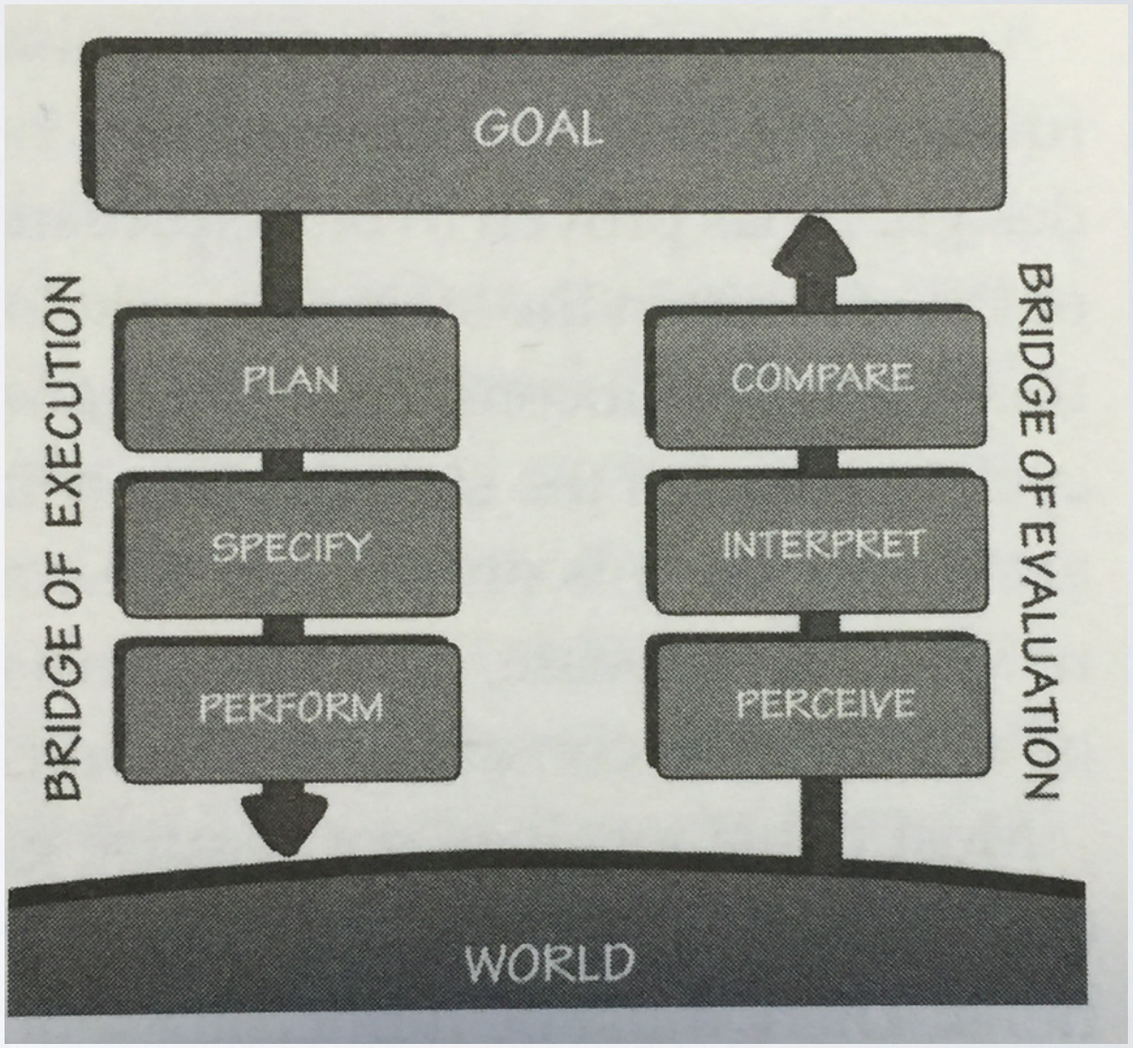
- Important users do not feel constrained
- Want users to feel that they can do things the way they want to do them, not as software dictates to them



Direct manipulation

Motivation

- User is trying to do a task, manipulating some [model] of world
- Hard to plan out long sequence of actions in advance
- Gulf of execution: hard to know if took correct action
- Gulf of evaluation: hard to understand if successfully manipulated world
- Hard to compare hidden world to desired world



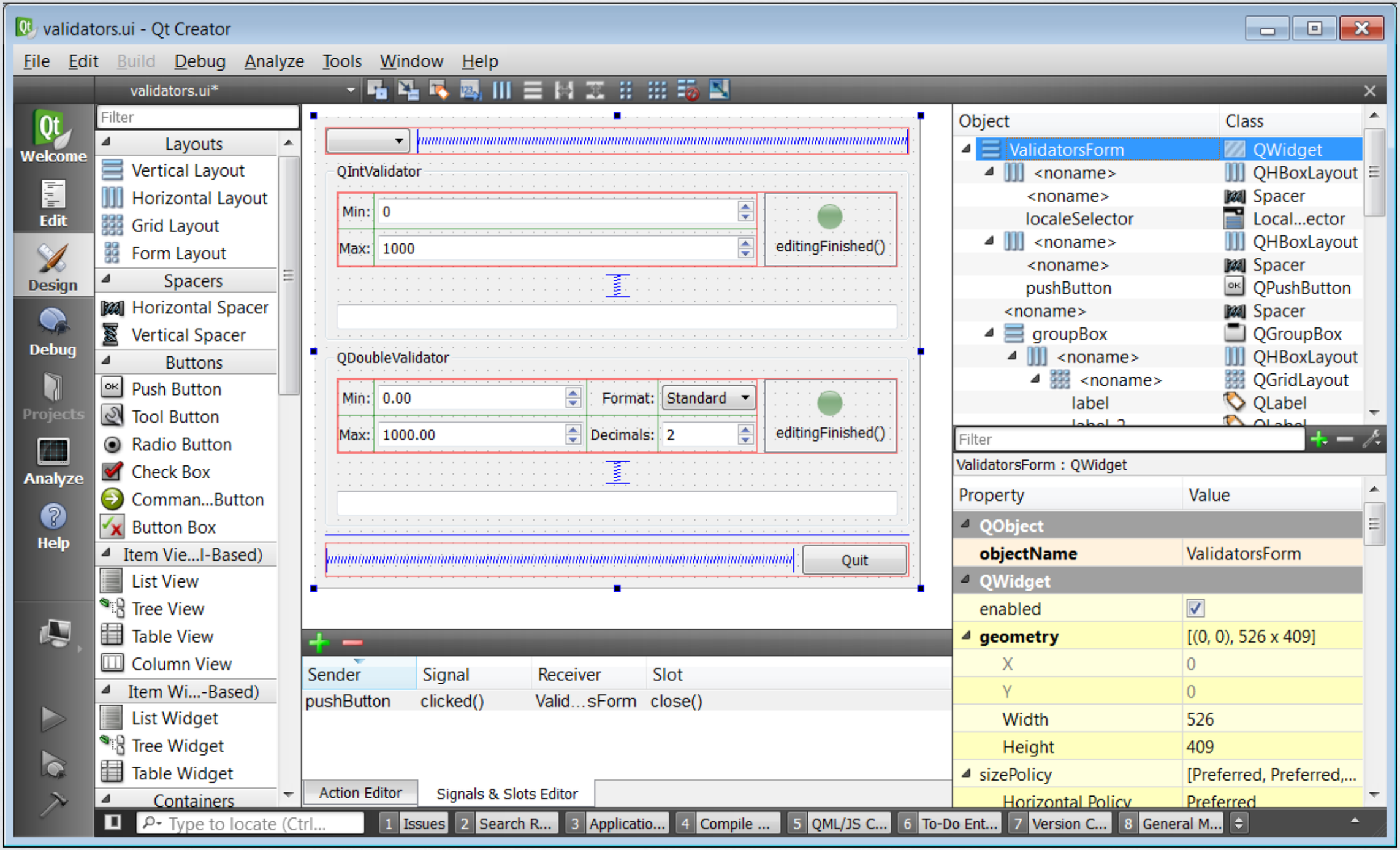
Direct manipulation

- "Rapid incremental reversible operations whose impact on the objects of interest is immediately visible" (Shneiderman, 1982)

Benefits

- Supports exploration
  - Don't plan long sequence of actions: pick an action, try it, can change mind if want to do something else instead
- Provides immediate feedback
  - Can quickly see what outcome of actions are in manipulating the world
  - Easy to compare desired state of the world to actual state of the world

Example - GUI builder



Example - Spreadsheets

FlyCalc - WIG2004.XLS

File Edit View Insert Format Tools ?

Support Chat OFF

Arial 10

Formula :

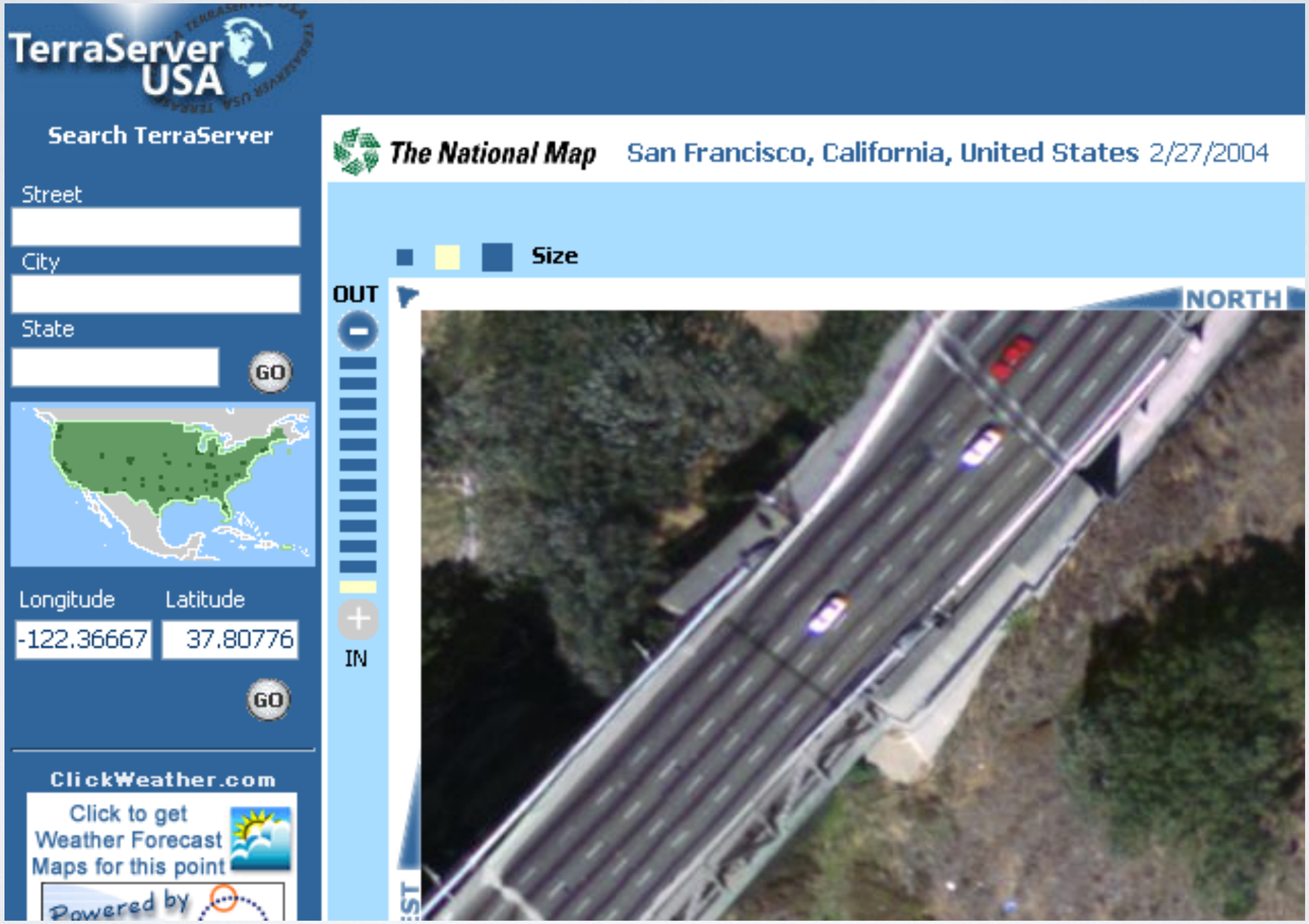
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		788	355	564	399	413	897	444	523	413						
2		800	923	233	307	864	355	90	877	864						
3		657	788	755	444	455	478	432	405	455						
4		599	866	233	201	413	361	455	233	413	Sep 2005	Oct 2005	Nov 2005	Dec 2005		Jan 2006	Feb 2006
5		899	755	673	311	780	400	614	754	780							
6			334	953	888	214	644	789	361	978							
7		233	644	766	446	231	977	577	453	847	455	507	690	700		788	800
8		577	533	968	897	541	977	475	358	975	355	478	361	400		355	923
9											742	267	599	700		564	233
10	Bryant Park	965	365	233	708	564	344	78	359		997	352	215	836		399	307
11	Keokuk	670	607	233	846	980	544	613	523		877	405	233	754		413	864
12	Westport	855	732	908	556	352	315	635	413		864	455	413	780		897	355
13	Temple	607	244	641	908	561	555	314	467		900	378	723	382		444	90
14	Lockhart	222	645	999	182	388	905	814	444		190	432	455	614		523	877
15	Stonington	344	756	600	481	339	489	144	399		307	444	201	311		413	864
16																	
17	Subtotal	5455	4380	5088	5002	4521	4866	4084	4342		5687	3718	3890	5477		4796	5313
18																	
19	**U.K. Factories**																
20																	
21	Clacton	855	315	908	556	352	556	635	413		864	455	413	780		980	966
22	Penge	506	605	860	222	459	222	521	897		355	478	361	400		670	800
23	Runcorn	670	544	233	846	980	846	613	523		877	405	233	754		2242	1543
24	Worcester Park	344	489	600	481	339	481	144	399		307	444	201	311		899	900
25	Wapping	855	315	908	556	352	556	635	413		864	455	413	780		600	650
26	Tooting Bec	506	605	860	222	459	222	521	897		355	478	361	400		600	670
27	Balham	222	905	999	182	388	182	814	444		90	432	455	614		797	668
28	Wigan	670	544	233	846	980	846	613	523		877	405	233	754		800	796
29	Ashby de la Zouche	855	315	908	556	352	556	635	413		864	455	413	780		413	780
30	Bude	607	555	641	908	561	908	314	467		900	378	723	382		361	400
31	Looe	344	489	600	481	339	481	144	399		307	444	201	311		455	614
32	Scunthorpe	674	677	790	650	666	679	677	566		756	567	685	433		900	780
33																	
34	Subtotal	5073	4761	5982	5078	4750	5078	4433	4478		5441	3896	3233	5086		7167	7021
35																	
36	**Canadian Factories**																
37																	
38	Deception Bay	344	489	600	600	481	339	521	897		355	478	361	233		846	613
39	Mississauqua	855	315	908	600	481	339	481	855		315	908	556	352		481	144
40	WIG																

FlyCalc 1.1 - Copyright Natium 2003-2006

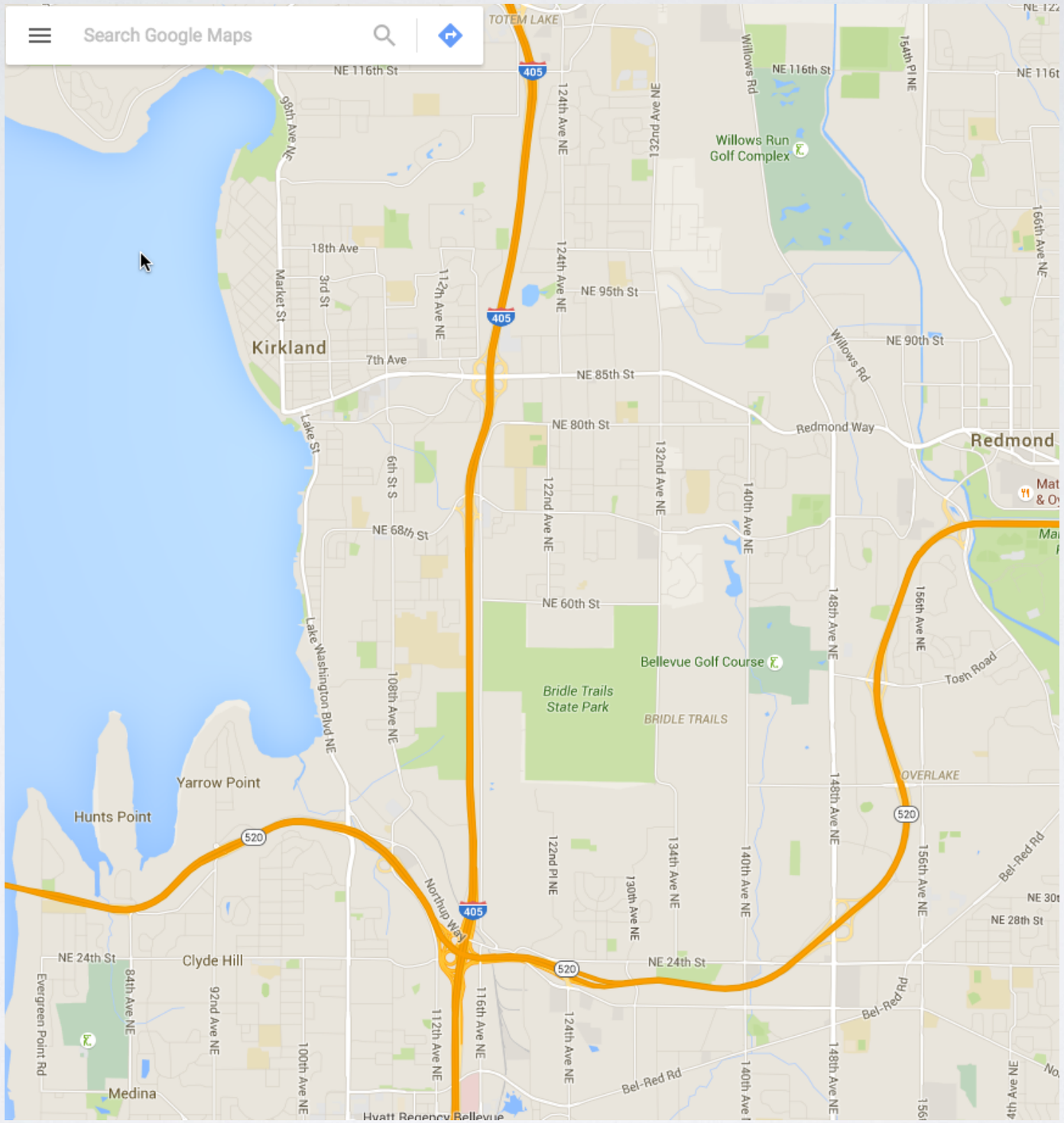
Example - Programming

- Bret Victor's Learnable Programming

Example - Microsoft TerraServer



Example - Google Maps



Physical actions

Provide intermediate feedback during interactions

- As user is interacting with objects, provide feedback on interactions

- Examples
  - While dragging object, show new position
  - As selecting text, show selection
  - While clicking on button, show button changing

Avoid physical awkwardness

- Switching between input devices takes time
- Avoid forcing user to constantly switch between input devices (e.g., keyboard & mouse)
  - e.g., Effective tab order between fields
- Avoid awkward keyboard combinations

Dimensions of user disabilities

- **Perception** - visual & auditory impairments
  - Blindness or visual impairments
  - Color blindness
  - Deafness & hearing limitations
- **Motion** - muscle control impairments
  - Difficulties with fine muscle control
  - Weakness & fatigue
- **Cognition** - difficulties with mental processes
  - Difficulties remembering
  - Difficulties with conceptualizing, planning, sequencing actions

Design for all

- How can users with physical disabilities be supported in user interactions?

- Good: **assistive design** - offering equivalent actions for disabled users that cannot take normal actions

- Better: **design for all** - designing interactions so broadest set of users across age, ability, status in life can use normal actions

Example - Curb cut

- Initially designed for **accessibility** - support for disabled & wheel chairs

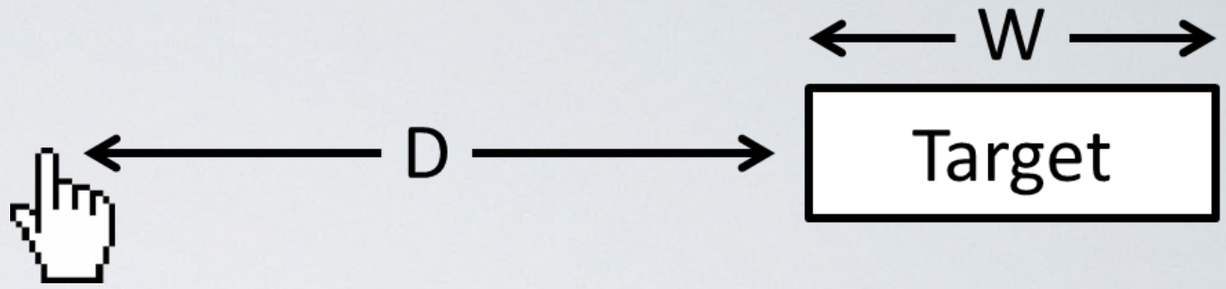
- But potentially benefits **all users** of public spaces - people w/ suitcases, hand carts, roller blades, bikes, …

7 Principles of Universal Design

- **Equitable use**: The design is useful and marketable to people with diverse abilities
- **Flexibility in use**: The design accommodates a wide range of individual preferences and abilities
- **Simple and intuitive**: Use of the design is easy to understand, regardless of the user's experience, knowledge, language skills, or current concentration level
- **Perceptible information**: The design communicates necessary information effectively to the user, regardless of ambient conditions or the user's sensory abilities
- **Tolerance for error:** The design minimizes hazards and the adverse consequences of accidental or unintended actions
- **Low physical effort**: The design can be used efficiently and comfortably and with a minimum of fatigue
- **Size and space for approach and use**: Appropriate size and space is provided for approach, reach, manipulation, and use regardless of user's body size, posture, or mobility

http://www.ncsu.edu/ncsu/design/cud/about_ud/udprinciples.htm

Fitt's law



- Time required to move to a target **decreases** with target **size** & **increases** with **distance** to the target

- Movements typical consist of
  - one large quick movement to target (**ballistic** movement)
  - fine-adjustment movement (**homing** movements)

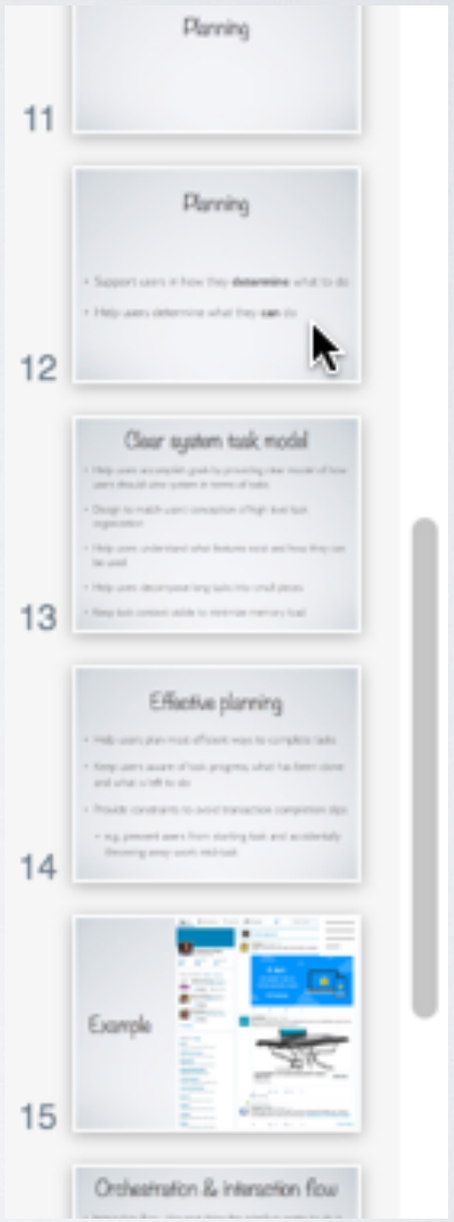
- Homing movements generally responsible for most of movement time & errors

- Applies to rapid pointing movements, not slow continuous movements

Design implications of Fitt's law

- **Constraining** movement to one dimension dramatics increases speed of actions

- e.g., scroll bars are 1D

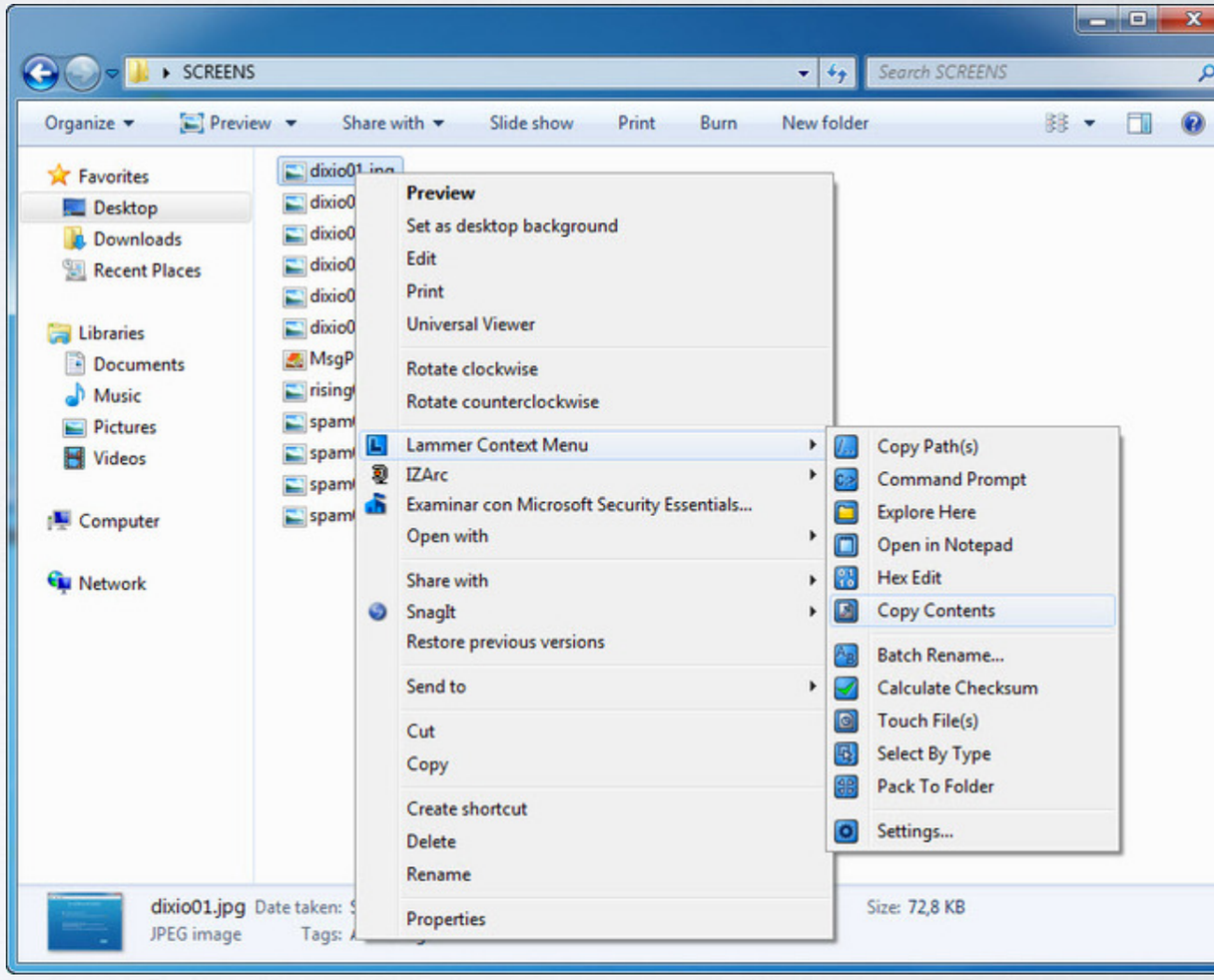


Design implications of Fitt's law

- Making controls **larger** reduces time to invoke actions

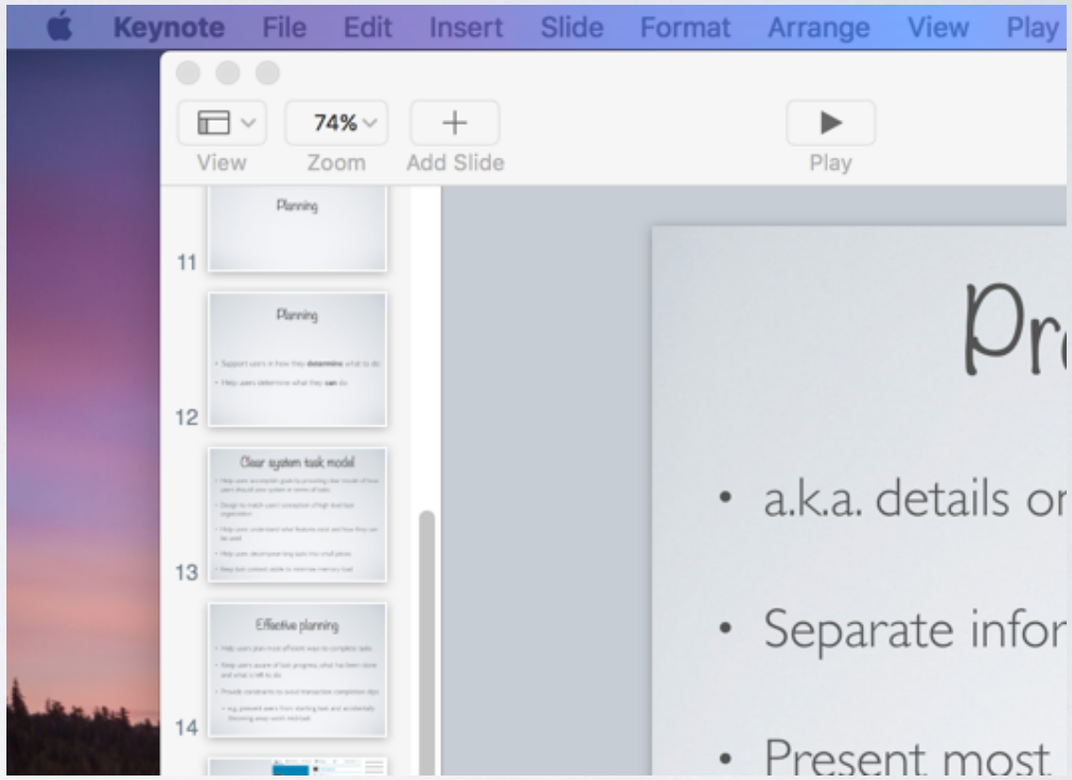
- Locating controls closer to user **cursor** reduces time

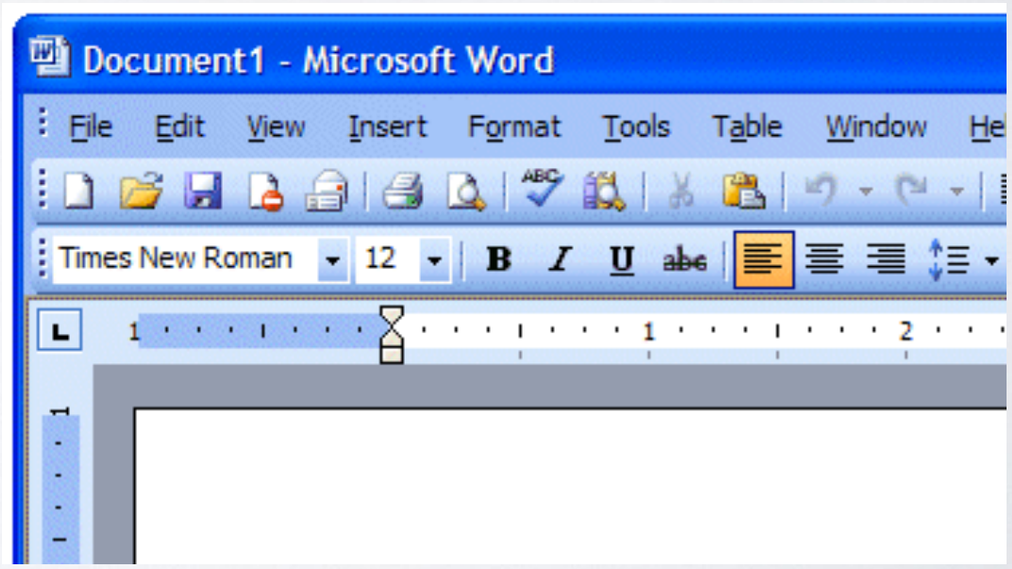
  - e.g., context menus



Design implications of Fitt's law

- Positioning button or control along **edge** of screen acts as barrier to movement, substantially reducing homing time & errors





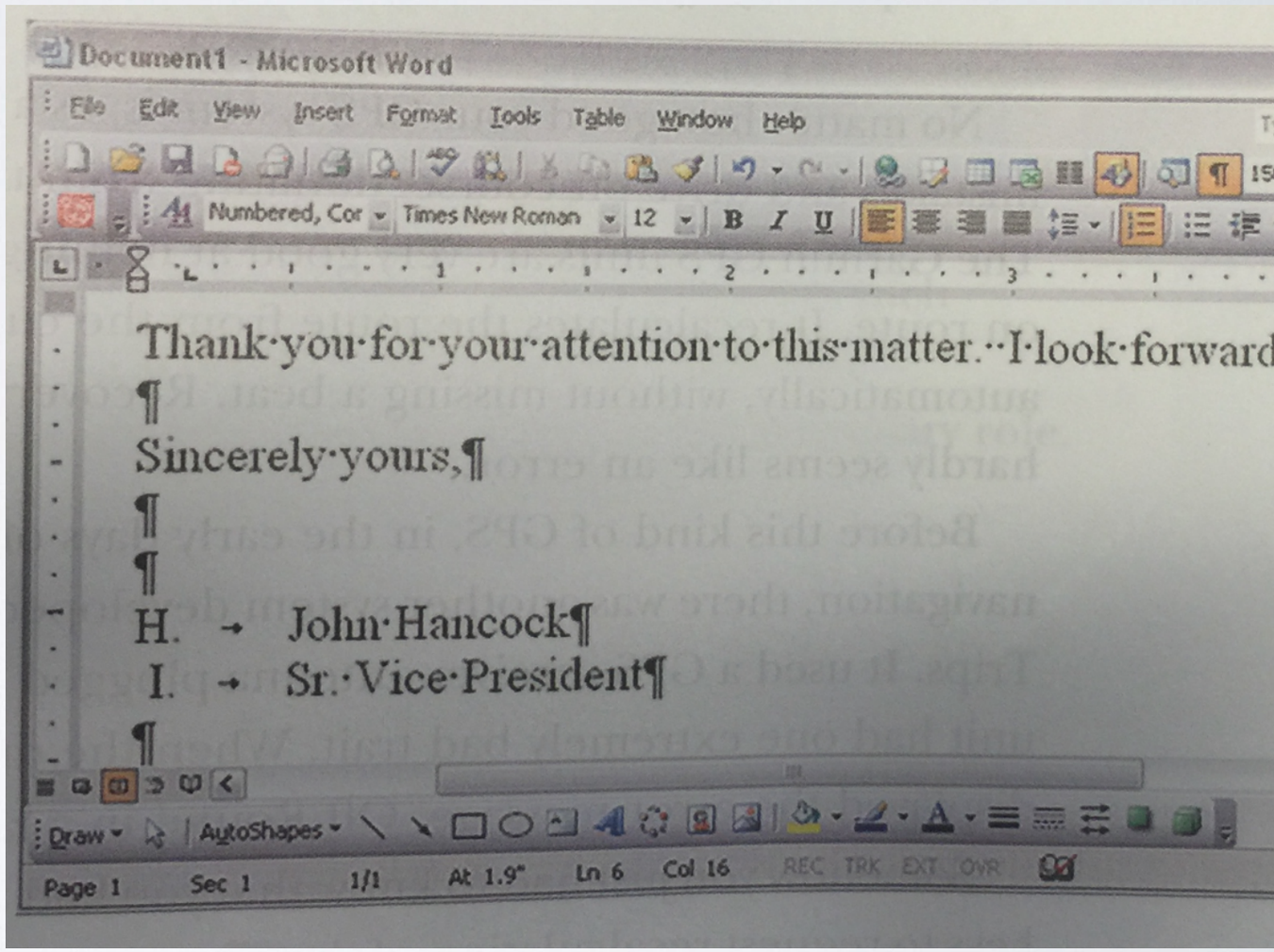
System feedback

System response times

- 0.1 second - reacting **instantaneously**
  - requiring no special feedback except displaying result
  - limit for direct manipulation of objects in UI
- 1.0 second - **freely** navigating commands
  - noticeable delay, limit for keeping user's flow of thought uninterrupted
- 10 seconds - keeping users **attention**
  - limit for keeping user's attention focus in UI
  - longer delays create task breaks
- [Nielsen, Usability Engineering, 1993]

Automation

- Keep user in control at highest task levels
- Take control from user when need is obvious & user is busy
- Provide visibility of automation & opportunities to correct when necessary

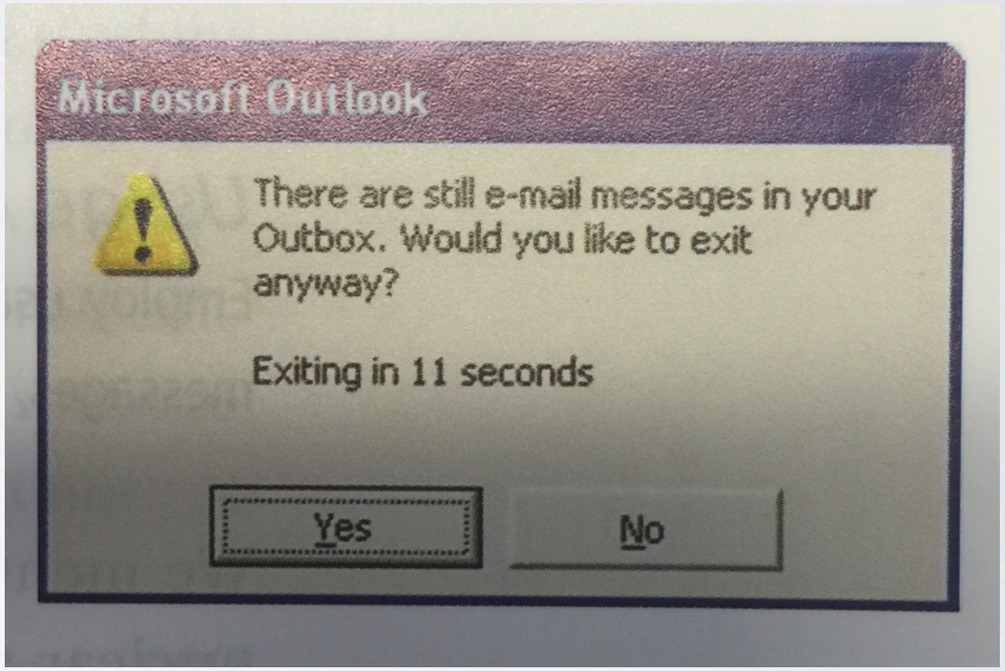


Provide feedback for all user actions

- Feedback helps keep users on track in accomplishing goals
- Request confirmation to prevent costly errors (but use sparingly)
- Make feedback visible, noticeable, legible, located w/ in users focus of attention
- Provide feedback early
- Provide feedback consistently

Crafting feedback text

- Clarity - support clear understanding of outcome
- Precise wording
- Completeness - include enough information to fully understand outcomes



Tone of feedback

- Establishes relationship with user
- Important not to take user feel "stupid"
- Make the system take blame for errors
- Be positive, to encourage
- Provide helpful messages, not cute messages
- Avoid violent, negative, demeaning, threatening terms (e.g., illegal, invalid)

Avoid anthropomorphism (in most contexts)

- Anthropomorphism - the attribution of human characteristics to non-human objects
  - e.g., "Sorry, I but I cannot find the file you need"
- Provides a false mental model
  - leads to user thinking they can interact with system as person
  - can be over promising & condescending
- May work in spoken interaction settings, where system does match user's mental model

In Class Activity

Interaction design critique

- In groups of 2
  - Pick an application or web app you know well
    - Should be something with long and complex user tasks (e.g., Photoshop, Illustrator, Word, Excel, Eclipse, VS)
- Critique the application from an interaction perspective
- Identify interaction strengths of the application
- Identify interaction weaknesses of the application