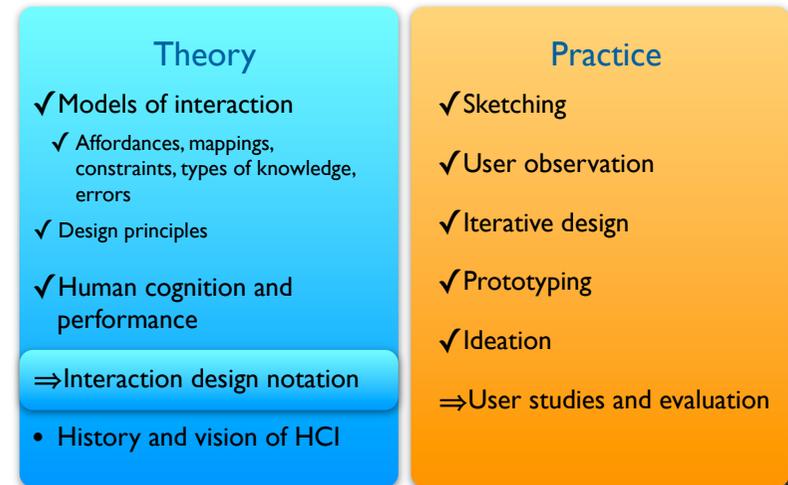


# Review

- What are two measures of interface efficiency?
  - How do you calculate them?
- Given  $n$  alternatives with different probabilities  $p(i)$ , what's the formula to compute the information per alternative?
- What are *gestures* in KLM?
- What are the limitations of GOMS?
- Techniques for qualitative evaluation with users?
- What document should users sign before your study?



# Back to BASICS

```
Print "Please enter a number"
```

```
INPUT n
```

```
Print "The square of",n,"is",n*n
```

- What are the problems with using such a notation to specify a dialog?

# Why UI Specification Languages

- In normal programming languages, UI and algorithms are mixed up
- System and user decisions are hard to distinguish
- Error checking on inputs dominates and complicates code
- First step: bundling I/O in classes/procedures
- Second step: Use a more efficient, readable language to specify the dialog
  - A priori to design the dialog
  - As part of the implementation (executable spec.)



# Specifying User Interfaces

- Problem: Describe the proposed design of a user interface
- Approach: natural/semi-formal/formal languages
- Many standard computer science techniques apply
- The more modern the UI, the harder to describe textually, depending on modality and UI style



# Grammars

- Mostly BNF-like

```
expr ::= empty
      | atom expr
      | '(' expr ')'
```

- E.g., Shneiderman's multiparty grammar

```
<Session> ::= <U: Opening> <C: Responding>
<U: Opening> ::= LOGIN <U: Name>
<U: Name> ::= <U: string>
<C: Responding> ::= HELLO [<U: Name>]
```

- Great for command-line UIs, e.g., banking ATMs, Unix commands
- Less suitable for GUIs



# Grammars

- Regular expressions

select-line click click\* double-click

- E.g., Unix “copy” command synopsis:

```
cp [-R [-H | -L | -P]] [-f | -i | -n] [-pv] source_file target_file
cp [-R [-H | -L | -P]] [-f | -i | -n] [-pv] source_file ... target_dir
```

recursion policies      overwrite policies

- Short and precise, but hard to read, requires additional information about semantics



# Production Rules

- Unordered list of rules: 

if <i>condition</i> then <i>action</i>
--

- Condition based on state or pending events
- Every rule always potentially active

- Good for concurrency
- Bad for sequence



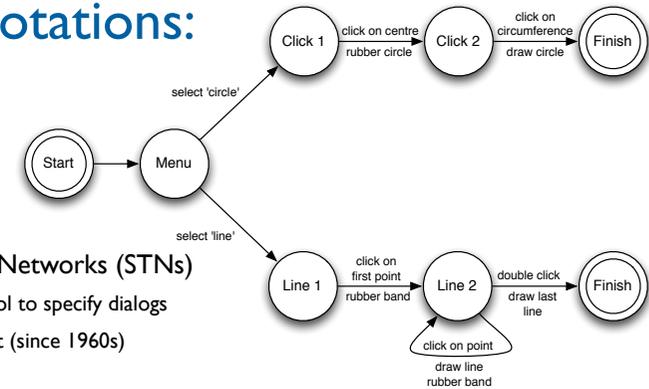
# Event Based Production Rules

select-line → first  
 click first → rest  
 click rest → rest  
 double-click rest → < draw line >

- Note:
  - Events added to list of pending events
  - 'first' and 'rest' are internally generated events
- Bad at state!



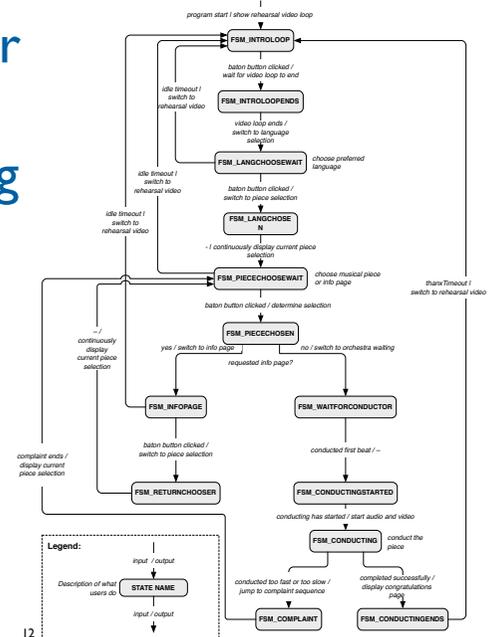
# Graph Notations: STNs

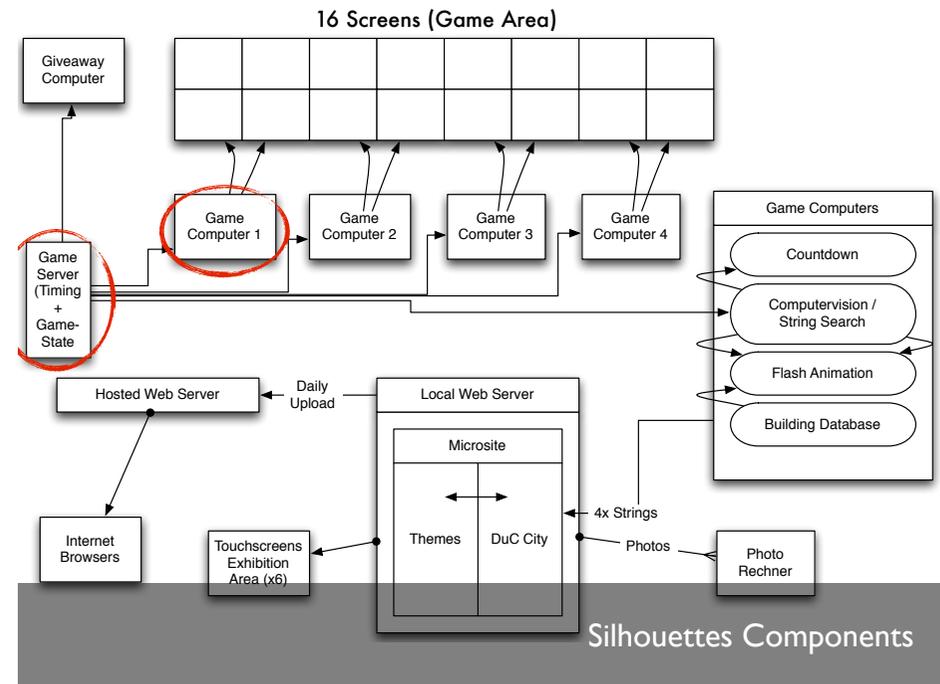
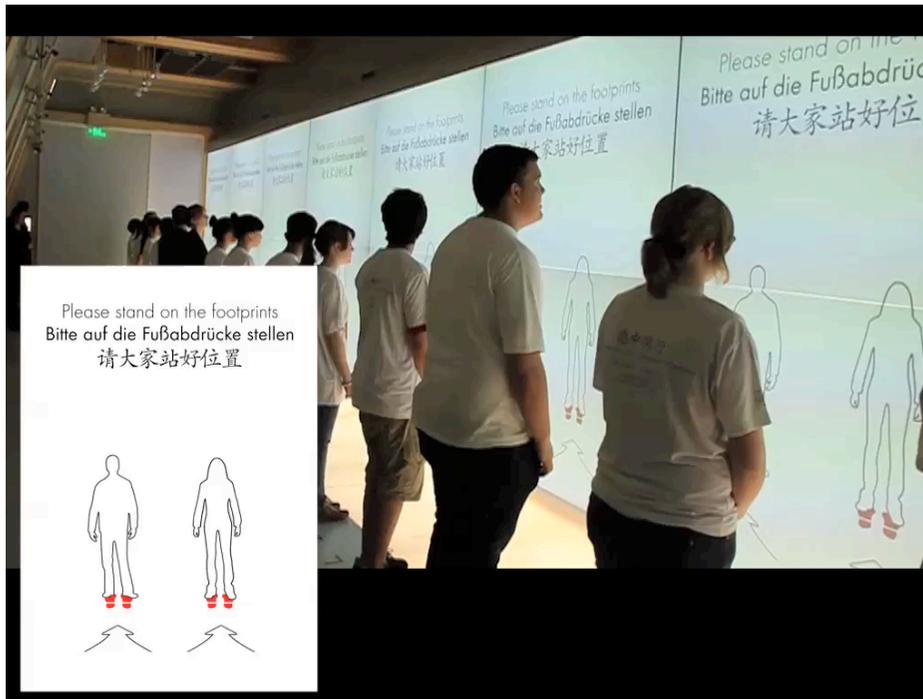
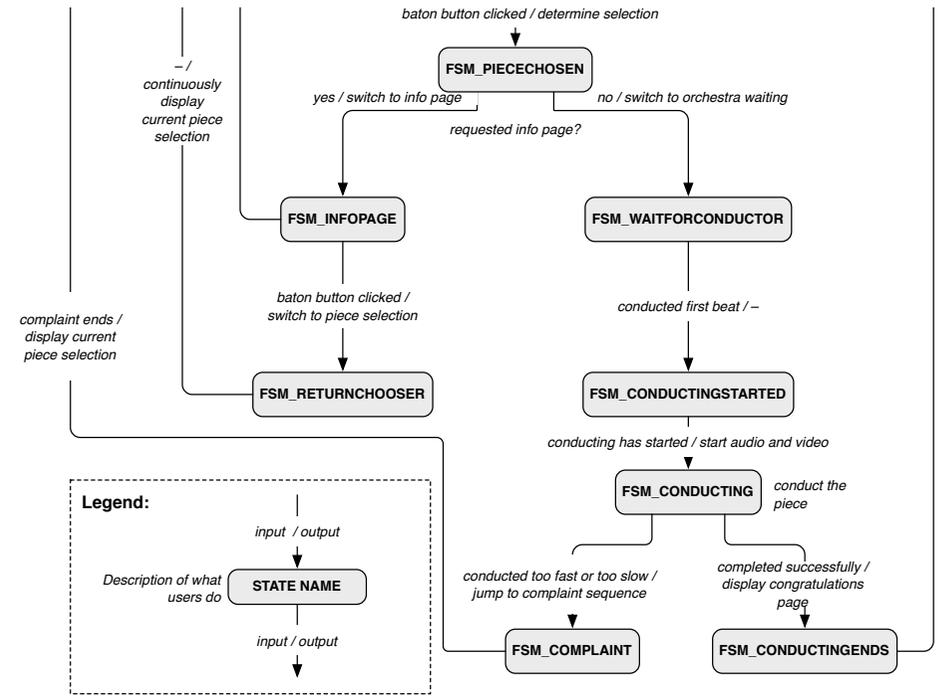
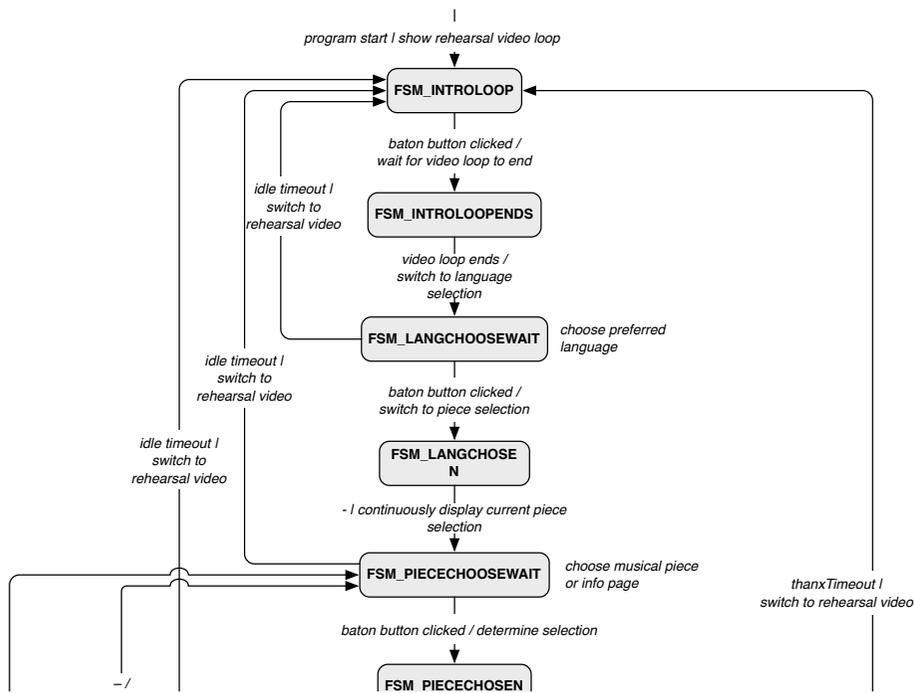


- State Transition Networks (STNs)
  - Most common tool to specify dialogs
  - Established format (since 1960s)
- Consisting of:
  - States (usually the system waiting for some user action)
  - Transitions (which have a user action and a system response associated with them)
- Describes sequences of user actions and system responses

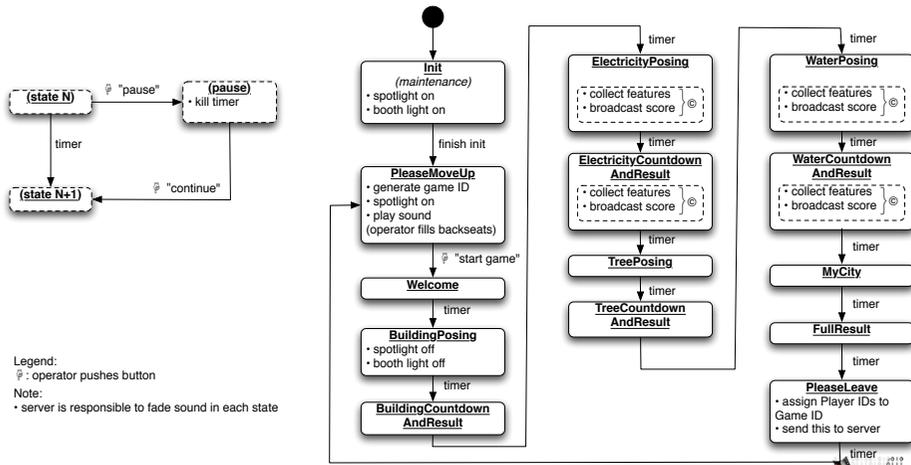


# Example: STN for Personal Orchestra Dialog





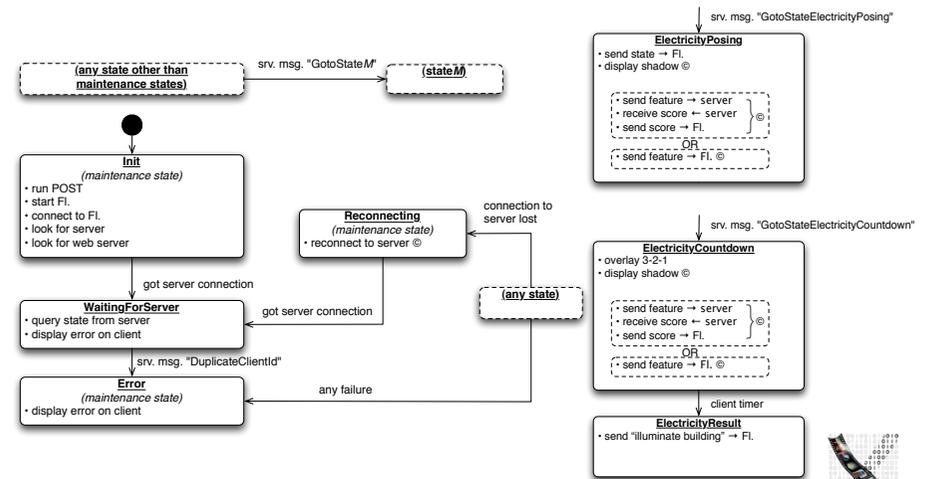
# Silhouettes: Server STN



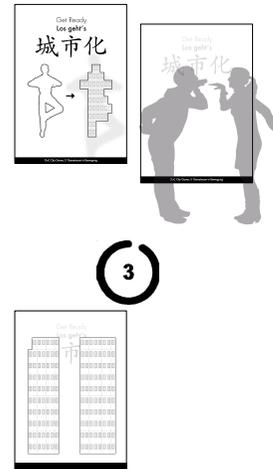
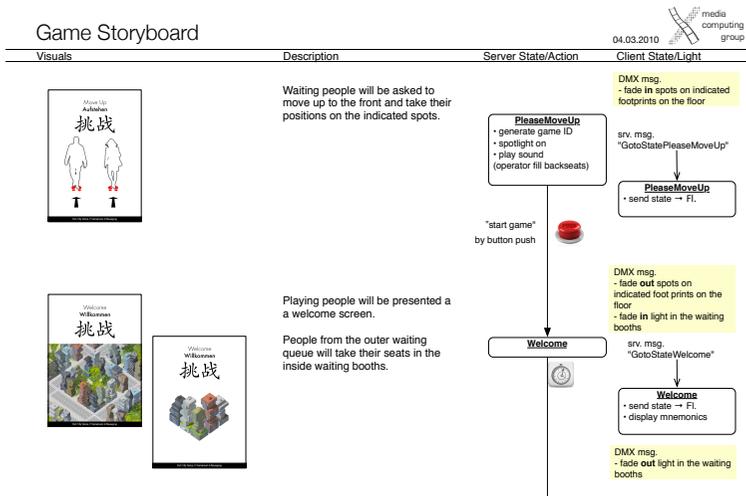
Legend:  
 ⏸: operator pushes button  
 Note:  
 • server is responsible to fade sound in each state

• Unconventional notation (agreed upon in the team)

# Silhouettes: Client STN



# Storyboard Meets STN

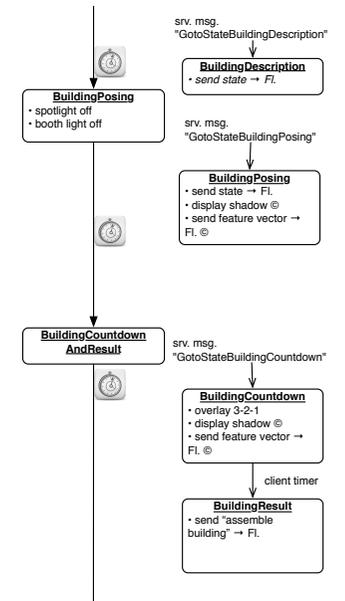


Iconic illustrations will remind the player how to create buildings by their shadows.

People will see their shadows while posing in front of the screens.

A countdown will indicate that shadows will be "frozen".

People will see the representation of their shadows as buildings.



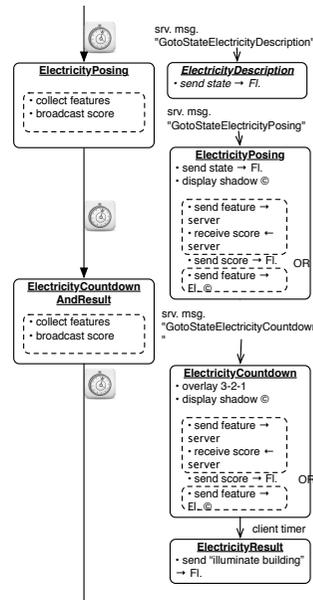


Iconic illustrations will remind the player how to pose in order to have their shadows overlap and thus create a power line.  
People will see their shadows while posing in front of the screens.

3



A countdown will indicate that shadows will be "frozen".  
People will see illuminated buildings in 4 different versions??



Iconic illustrations will remind the player how to create trees by their shadows.

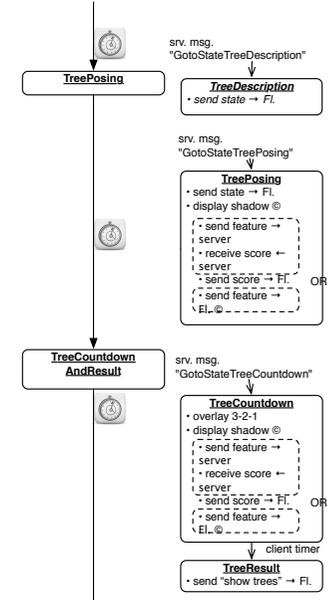
People will see their shadows while posing in front of the screens.

3



A countdown will indicate that shadows will be "frozen".

People will see the representation of their shadows as trees.

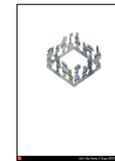
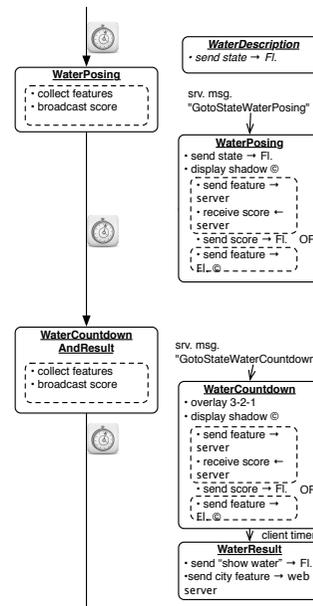


Iconic illustrations will remind the player how to pose in order to have their shadows overlap and thus create a water line.  
People will see their shadows while posing in front of the screens.

3



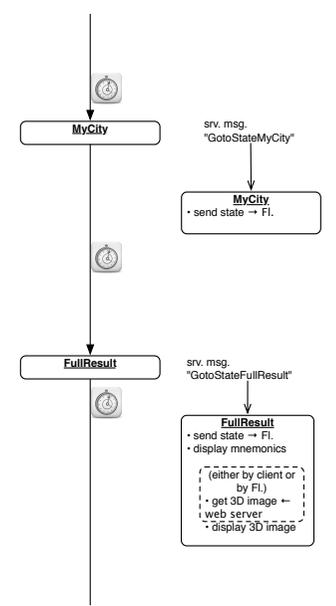
A countdown will indicate that shadows will be "frozen".  
People will see the trees that they created with different green colours/trees with leaves?



People will see an animation of what will happen with their skyline.

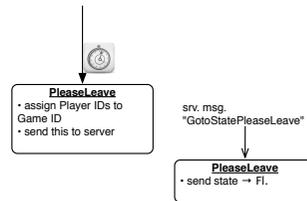


People will see their city block with the block number and the URL of the DuC Website.

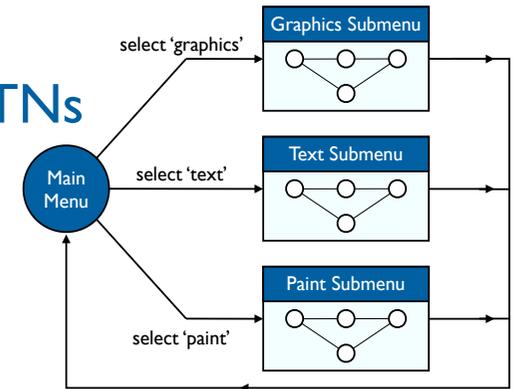




People will be asked to move out.  
Beyond the playing area, visitors can deepen their understanding of city concepts at individual interactive stations before leaving the pavilion.



## Hierarchical STNs



- **Start** and **Finish** states serve to glue an STN for a subdialog (e.g., a certain menu selection) into a larger dialog (e.g., operating the application in general)
- Same expressive power as STNs, just more convenient
- The dialog structure of an entire system can be specified this way



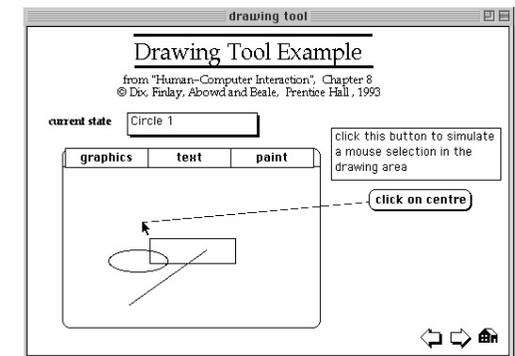
## Using STNs In Prototyping

- Create a simple STN for the dialogs envisioned
- Create one UI snapshot (sketch if paper prototype) per **state** (label it with the state name)
- Include offscreen area for annotations and to include extra buttons simulating user actions that do not correspond to simple clicks on the current screen
- When walking the user through your paper prototype, consult the STN to find out how to respond to each user action



## Using STNs In Prototyping

- Alternative: Let the computer “execute” the STN to run the prototype
- Use tools such as HyperCard, PowerPoint, Keynote
- Demo: Drawing tool prototype in HyperCard



INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

**Rankings**

.....  
mehr...

**Veranstaltungen**

.....  
Veranstaltung 1

.....  
Veranstaltung 2

.....  
Veranstaltung 3

.....  
mehr...

**Aktuelles**

.....  
Aktuelles 1

.....  
Aktuelles 2

.....  
Aktuelles 3

.....  
mehr...

**Video**

.....  
mehr...

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

mmm.2011 Abend der Offenen Tür  
 mmm.2011 Girls' Day  
 mmm.2011 Tag Der Informatik

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

Warum Informatik an der RWTH?  
 Girls' Day 2011  
 Schülerinformationstag 2011  
 Helle Köpfe 2011 für Grundschüler  
 5 vor 12: Die Wissenschaftsnacht 2011  
 Fit für Informatik? Mach' den Test!  
 Vorkurs Informatik

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

Bachelor Informatik  
 Master Informatik  
 Master of Science in Software Systems Engineering  
 Master of Science in Media Informatics  
 Promotionsstudium Informatik  
 Diplomstudiengang Informatik  
 Lehramtsstudiengang Informatik - Gymnasium und Gesamtschule  
 Schwerpunkt Informatik im Studiengang Technik-Kommunikation

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

Fachschaft Mathematik/Physik/Informatik

Studienberatung Bachelor Informatik

Studienberatung Master Informatik

Studienberatung Master Software Systems Engineering

Studienberatung Master Media Informatics

Studienberatung Lehramt Informatik

Studienberatung Technik-Kommunikation

Auslandsstudienberatung

Prüfungsausschüsse...

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

Informatik-Bibliothek

Rechnerbetrieb Informatik (RBI)

RWTH-Rechenzentrum

Alle Lehrveranstaltungen in CAMPUS

Das RWTH-Lernportal L2P

Vorlesungen auf iTunes U

Prüfungsordnungen

Formulare

Semestertermine

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Studieren bei uns?
Studiengänge
Ansprechpartner
Service
Absolventen

Master

Promotion

Berufseinstieg

Career Center

Alumni Office

INFORMATIK RWTHAACHEN Anfahrt Kontakt Impressum facebook twitter RSS Login English

Suche:

Studium
Forschung
Wirtschaft
Die Fachgruppe

Aktuelles
Personen
Gruppen
Projekte
Publikationen
Auszeichnungen
Forschen an der RWTH

...

...

...

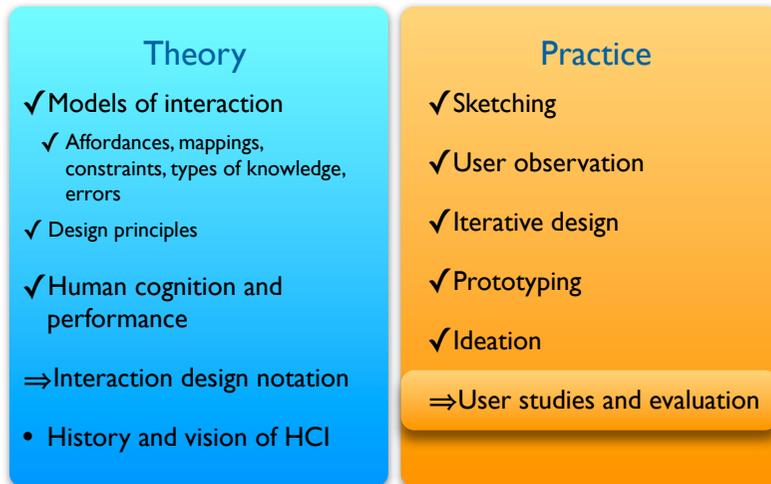
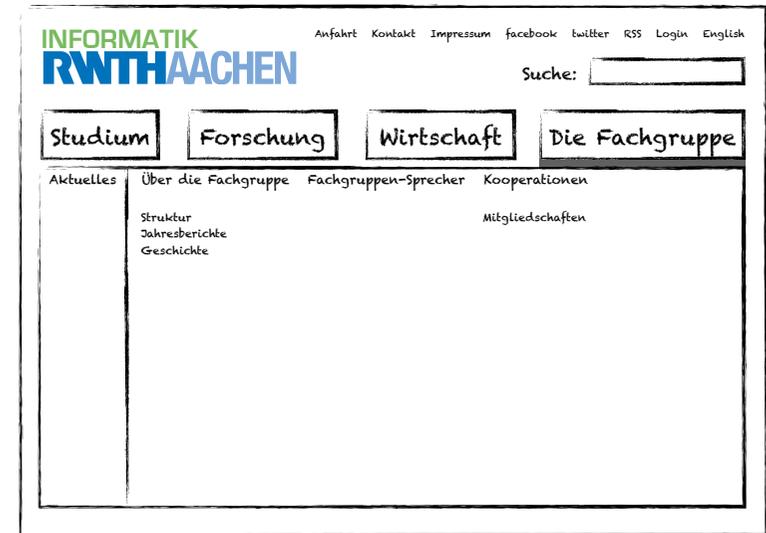
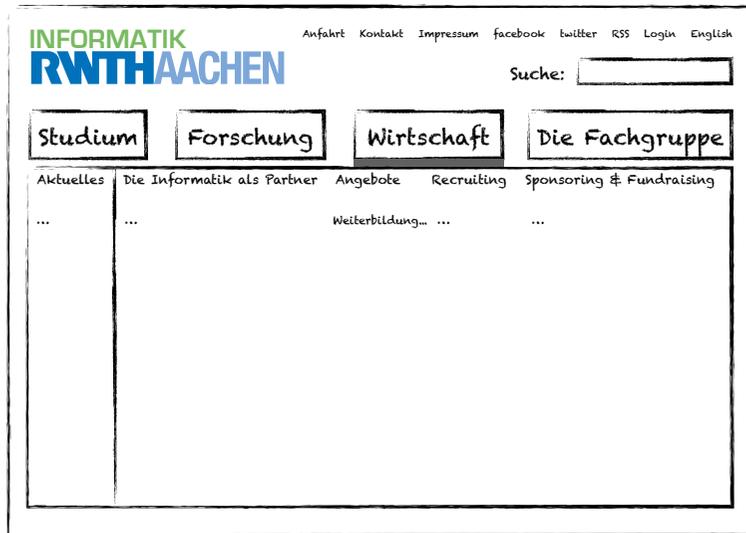
...

Presse...

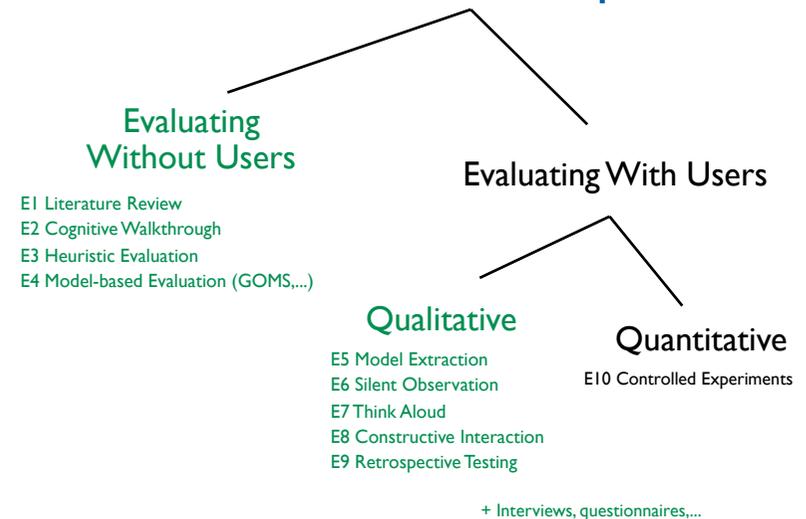
...

Stellenangebote

...

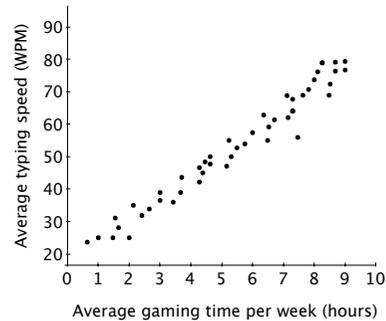


## Evaluation Techniques



# E10: Controlled Experiments

- Quantitative, empirical method
- Used to identify the cause of a situation or set of events
  - “X is responsible for Y”
  - Directly manipulate and control variables
- Correlation does not imply causality
  - Example: relationship between typing speed and time spent playing games
- Use a controlled experiment to verify an observation, a correlation, or a “hunch”



# Controlled Experiments: Steps

1. Formulate hypothesis
  - Pie menu decreases seek time for menu items
2. Design experiment, pick variable and fixed parameters
  - Type of menu  $\Rightarrow$  target seek time
3. Run a pilot study to debug your experimental procedures
  - Improving distribution of menu targets

The examples are simplified from [Schneiderman et al., CHI'88]



# Controlled Experiments: Steps

4. Recruit subjects
  - Undergraduate students with little or no mouse experience
5. Run experiment
  - Each participant perform menu selection 10 times for each type of menu
6. Interpret results to accept or reject hypothesis
  - Mean seek time: 2.26 (Pie), 2.64 (Linear)
  - The difference is statistically significant ( $p = .0003$ )

The examples are simplified from [Schneiderman et al., CHI'88]



# Controlled Experiments

- Subjects
  - Similar to real users in profile (age, education, computer and domain expertise, system knowledge, ...)
  - Use at least 10 subjects
    - Use more if you need finer details
    - Statistical power analysis can tell you the exact number
- Variables
  - **Independent Variables (IVs):** are varied under your control
    - E.g., number of menu entries
    - Each level of an independent variable is called a **treatment**
  - **Dependent Variables (DVs):** are those you measure
    - E.g., execution time, error rates, subjective preferences



# In-Class Exercise: Identifying Variables

- Identify independent variables and dependent variables from each of the following scenarios. Indicate levels of each independent variable:
  1. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training
  2. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets
  3. A research team examining whether virtual teams who use video instant messaging (IM) are more productive than teams who use text-only IM

Examples from: Research Methods in HCI, Lazar et al. (2010)



# Hypothesis

- Predicts outcome of experiment
  - Usually: claims that changing independent variables influences dependent variables
- Experiment goal: confirm **research hypothesis ( $H_1$ )**
  - “No amount of experimentation can ever prove me right; a single experiment can prove me wrong.”—Albert Einstein
- Approach: Reject inverse **null hypothesis ( $H_0$ )**, i.e., “no influence”
  - If we can determine that  $H_0$  is wrong, we can accept that  $H_1$  is true (naive view)
  - $H_0$  is usually a precise statement  $\Rightarrow$  we'll know the probability that  $H_0$  is incorrect
    - E.g., “Average WPM between gaming and non-gaming groups are equal”
  - The data should indicate that there is a very low probability that  $H_0$  is correct
- Being unable to reject  $H_0$  **does not imply** that you can accept  $H_0$ 
  - E.g., your number of participants may just have been too small



# In-Class Exercise: Identifying Hypotheses

- Identify a **research hypothesis ( $H_1$ )** and **null hypothesis ( $H_0$ )** for each of our scenarios:
  1. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training
  2. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets
  3. A research team examining whether virtual teams who use video instant messaging (IM) are more productive than teams who use text-only IM

Examples from: Research Methods in HCI, Lazar et al. (2010)



# Basic Experimental Designs

- **Between-groups design**
  - Each subject only does one variant of the experiment
  - There are at least 2 groups to isolate effect of manipulation:
    - Treatment group and control group
  - + No learning effects across variants
    - Good for tasks that are simple and involve limited cognitive processes, e.g., tapping, dragging, or visual search
  - But: requires more users
- **Within-groups design**
  - Each subject does all variants of the experiment
  - + Fewer users required, individual differences canceled out
    - Good for complex tasks, e.g., typing, reading, composition, problem solving
  - But: learning effects may occur



# In-Class Exercise: Experimental Designs

- Which type of experimental design is appropriate for each scenario?
  1. A study investigating whether people who have attended a security training program generate and use more secure passwords than people who haven't received any security training
  2. A research team examining the effectiveness of joysticks and trackballs for selecting static and moving targets
  3. A research team examining whether virtual teams who use video instant messaging (IM) are more productive than teams who use text-only IM

Examples from: Research Methods in HCI, Lazar et al. (2010)



# Within-Group Design: Order Effect

- The order of presenting the treatments (IV levels) might affect the dependent variable
  - Learning effect
  - Fatigue effect
  - Contrast effect: the effect of the first treatment carries over to influence the response to the second treatment
- Solutions
  - Rest period between treatments
  - **Counterbalancing**: all possible orders of treatments are included — but:  $O(n!)$
  - **Latin Square**: A limited set of orders,  $O(n)$



# Latin Square

- Each condition appears at each ordinal position
- Each condition precedes and follows each condition one time
- Example: six treatments: A, B, C, D, E, F

1	A	B	F	C	E	D
2	B	C	A	D	F	E
3	C	D	B	E	A	F
4	D	E	C	F	B	A
5	E	F	D	A	C	B
6	F	A	E	B	D	C



# Randomization

- Randomly assign treatment to participants
- Prevents systematic bias
- But: randomization  $\neq$  counterbalancing
  - With small numbers, randomization might not cover all combinations



# Analyzing Results

More detail: next lecture

- Do statistical analysis using well-defined test methods
  - E.g., Student's  $t$ -test, ANOVA (analysis of variance), regression analysis, Wilcoxon- or Mann/Whitney test,  $\chi^2$  test
- Choice depends on number, continuity, and assumed distribution of variables, and the desired form of the result
  - Results can be simple “yes/no”, size of difference, or confidence of estimate



# Making Your Evaluation Valid and Reliable

- **Validity:** How accurate is your result?
- **Reliability:** How consistent or stable is your result?
  
- These apply to *all* evaluations — not just controlled experiments



## Validity

- **Construct validity:** Were variables defined concretely enough to be manipulated or measured?
  - ✗ Speed
  - ✓ Time since the cursor leaves the start until when it reaches the target
- **Internal validity:** Is the causal inference logical? How strong is it?
  - Usually higher in experimental methods than descriptive or correlational methods
- **External validity:** Can the result be generalized to other populations and settings?
  - Evaluations in the lab usually have lower external validity than those in the field
- **Conclusion validity:** Were the conclusions from the data reasonable?
  - Is the result significant enough to say that there is a relationship between variables (without concerning causality)?

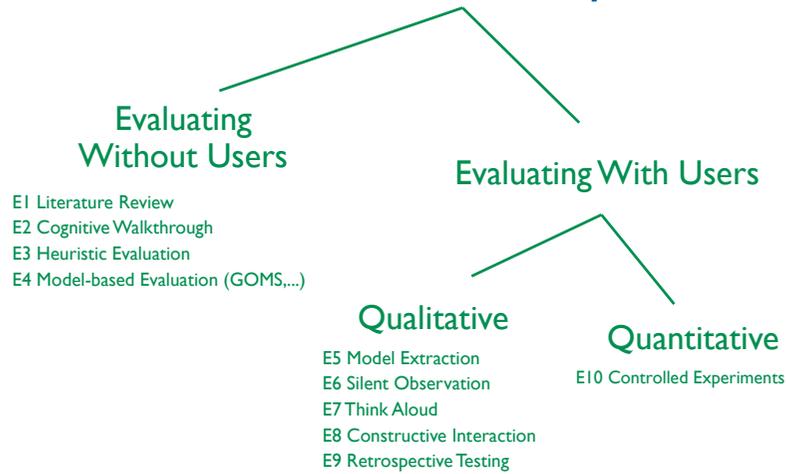


## Reliability

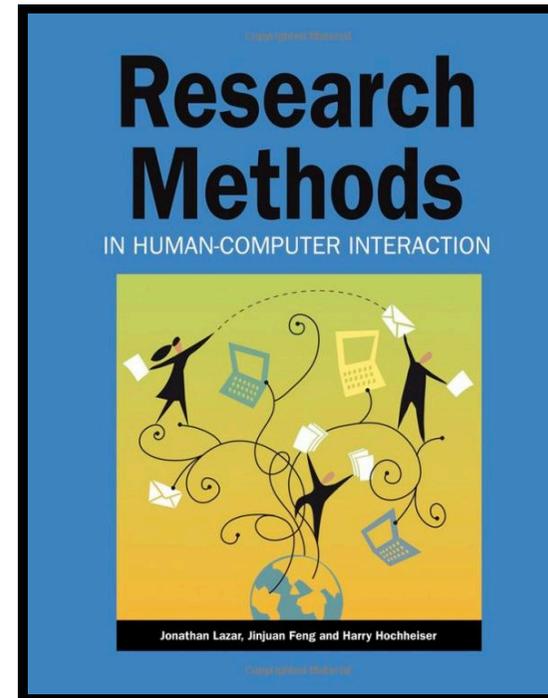
- Can the experiment be **replicated** by other research teams in other locations and yield results that are consistent, dependable, and stable?
- Does the experimental procedure clearly described in the paper or the report?
- Other causes of fluctuation
  - **Random errors:** cannot be eliminated — testing with more samples can help
  - **Systematic errors:** push the measured value into the same direction, caused by:
    - Measurement instruments
    - Experimental procedures: not randomized, not counterbalanced, instructions are biased
    - Participants: the recruitment process may filter participants unevenly
    - Experimenter behavior: bias in spoken language during experiment
    - Environmental factors: physical environment might favor one treatment over others



# Evaluation Techniques



+ Interviews, questionnaires,...



Recommended reading for more details about evaluation methods — especially if you are going to do your thesis at our chair!