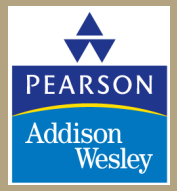
Lecture 14
Design Models 2 – GOMS and State Transition

Prof Jim Warren

with reference to sections 7.3 and 7.5 of

The Resonant Interface
HCI Foundations for Interaction Design
First Edition

by Steven Heim



Chapter 7 Interaction Design Models

- Model Human Processor (MHP)
- Keyboard Level Model (KLM)
- **GOMS**
- Modeling Structure
- **Modeling Dynamics**
- Physical Models

GOMS

Maxim

Goal/task models can be used to explore the methods people use to accomplish their goals

- Card et al. suggested that user interaction could be described by defining the sequential actions a person undertakes to accomplish a task.

- The GOMS model has four components:
  - goals
  - operators
  - methods
  - selection rules

GOMS

- **Goals** - Tasks are deconstructed as a set of goals and subgoals.
- **Operators** - Tasks can only be carried out by undertaking specific actions.
- **Methods** - Represent ways of achieving a goal
  - Comprised of operators that facilitate method completion
- **Selection Rules** - The method that the user chooses is determined by selection rules

GOMS – *CMN-GOMS*

MAXIM

CMN-GOMS can predict behavior and assess memory requirements

- CMN-GOMS (named after Card, Moran, and Newell) -a detailed expansion of the general GOMS model
  - Includes specific analysis procedures and notation descriptions
- Can judge memory requirements (the depth of the nested goal structures)
- Provides insight into user performance measures

CNM-GOMS example

```
GOAL: CLOSE-WINDOW
.   [select GOAL: USE-MENU-METHOD
.           .   MOVE-MOUSE-TO-FILE-MENU
.           .   PULL-DOWN-FILE-MENU
.           .   CLICK-OVER-CLOSE-OPTION
            GOAL: USE-CTRL-W-METHOD
.           .   PRESS-CONTROL-W-KEYS]

For a particular user, U1:

   Rule 1: Select USE-MENU-METHOD unless another
           rule applies
   Rule 2: If the application is GAME,
           select CTRL-W-METHOD
```

So here we have one Goal with either of two Methods, one of which requires a sequence of three

Operators, the other requires just one Operator; for U1 we have 2 Selection rules

GOMS – *Other GOMS Models*

- **NGOMSL** (Natural GOMS Language), developed by Kieras, provides a structured natural-language notation for GOMS analysis and describes the procedures for accomplishing that analysis
  - NGOMSL Provides:
    - A method for measuring the time it will take to learn specific method of operation
    - A way to determine the consistency of a design's methods of operation
- **Bonus learning** – see ftp://www.eecs.umich.edu/people/kieras/GOMS/NGOMSL_Guide.pdf

GOMS – *Other GOMS Models*

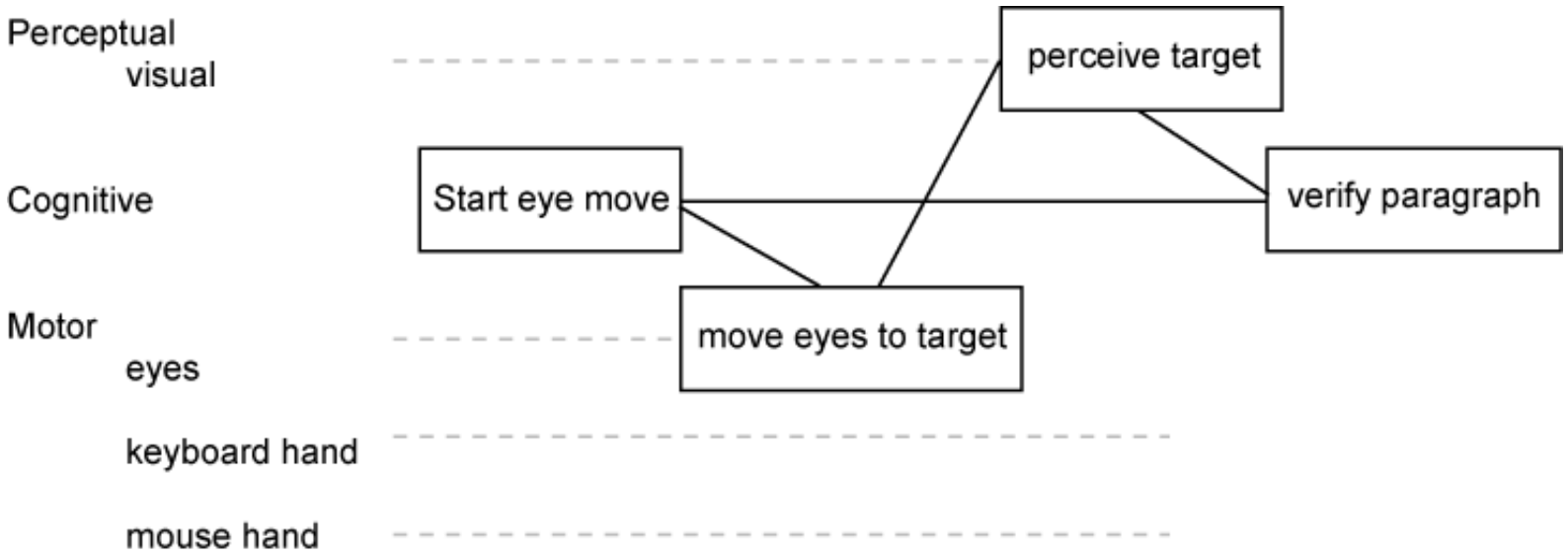
- **CPM-GOMS** represents
  - Cognitive
  - Perceptual
  - Motor operators

- **CPM-GOMS** uses Program Evaluation Review Technique (PERT) charts
  - Maps task durations using the critical path method (CPM).

- **CPM-GOMS** is based directly on the Model Human Processor
  - Assumes that perceptual, cognitive, and motor processors function in parallel

GOMS – *Other GOMS Models*

- Program Evaluation Review Technique (PERT) chart Resource Flows



Modeling Dynamics

MAXIM

Understanding the temporal aspects of interaction design is essential to the design of usable and useful systems

- Interaction designs involve dynamic feedback loops between the user and the system
  - User actions alter the state of the system, which in turn influences the user's subsequent actions
- Interaction designers need tools to explore how a system undergoes transitions from one state to the next

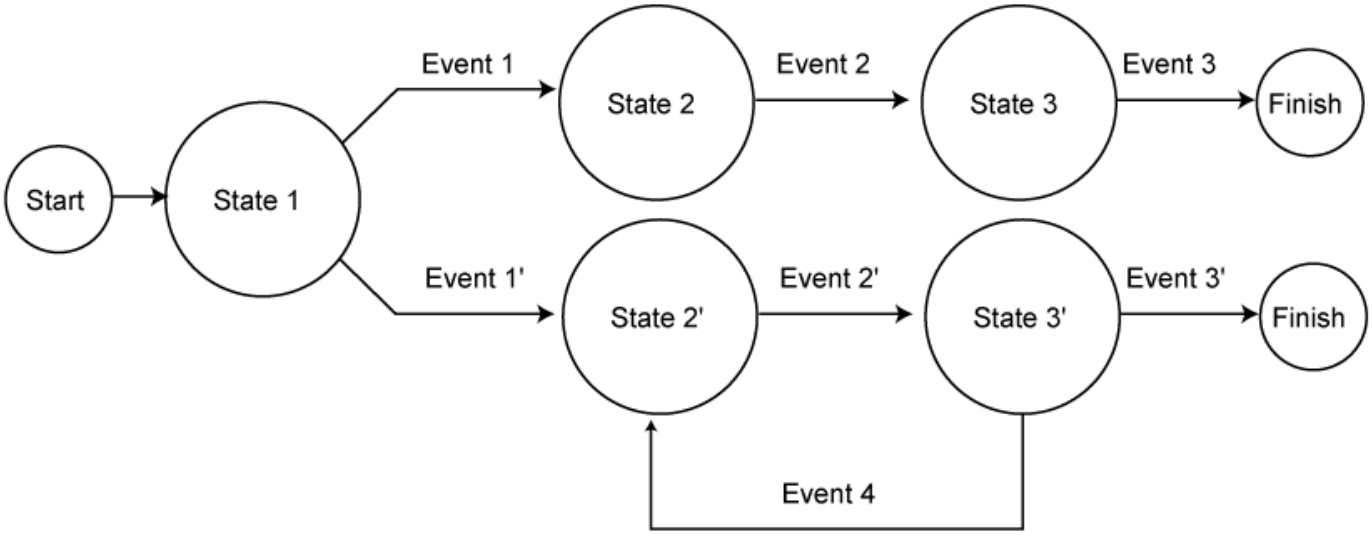
Modeling Dynamics – *State Transition Networks*

- **State Transition Networks** can be used to explore:
  - Menus
  - Icons
  - Tools

- **State Transition Networks** can show the operation of peripheral devices

Modeling Dynamics – *State Transition Networks*

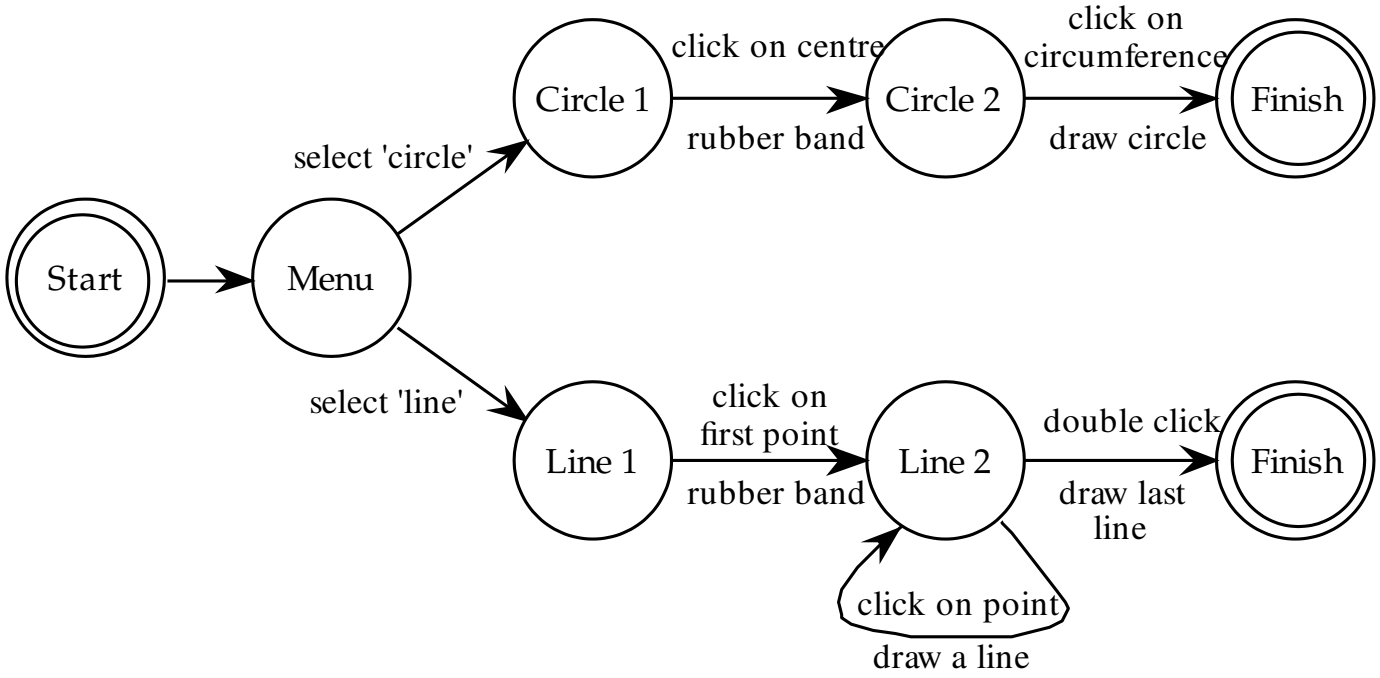
- State Transition Network



- STNs are appropriate for showing sequential operations that may involve choice on the part of the user, as well as for expressing iteration.

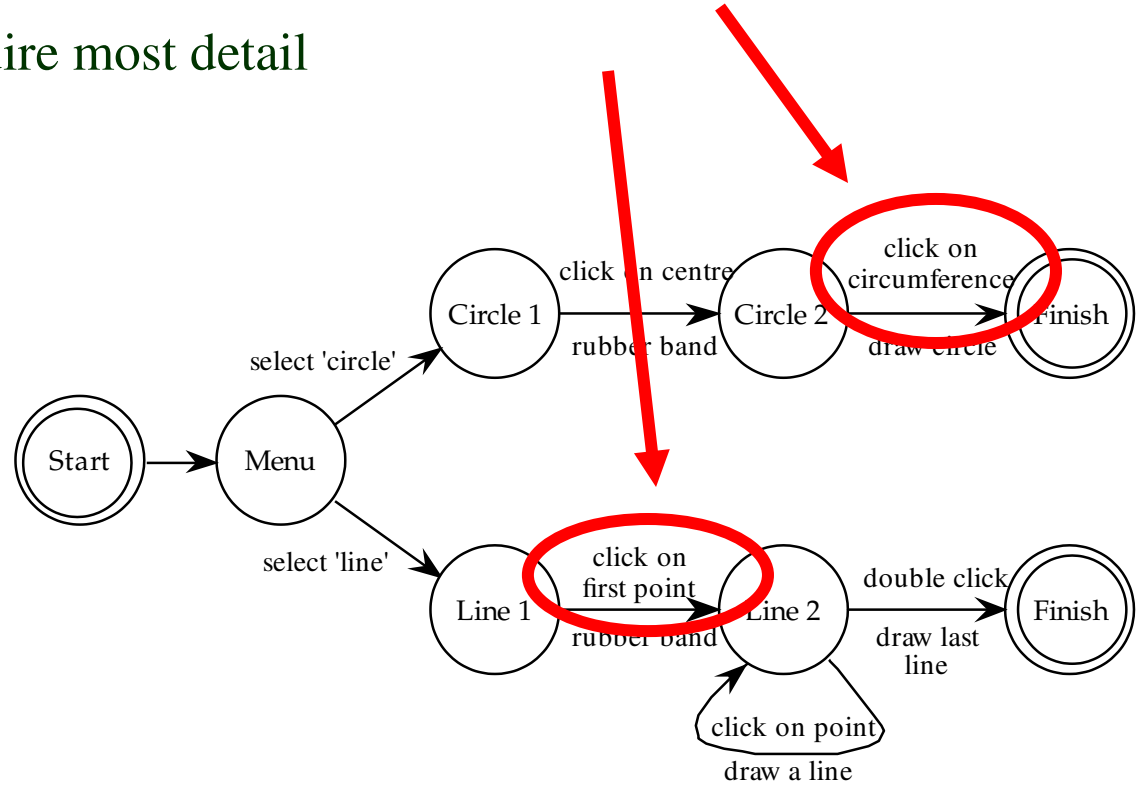
State transition networks (STN) – example

- circles - states
- arcs - actions/events



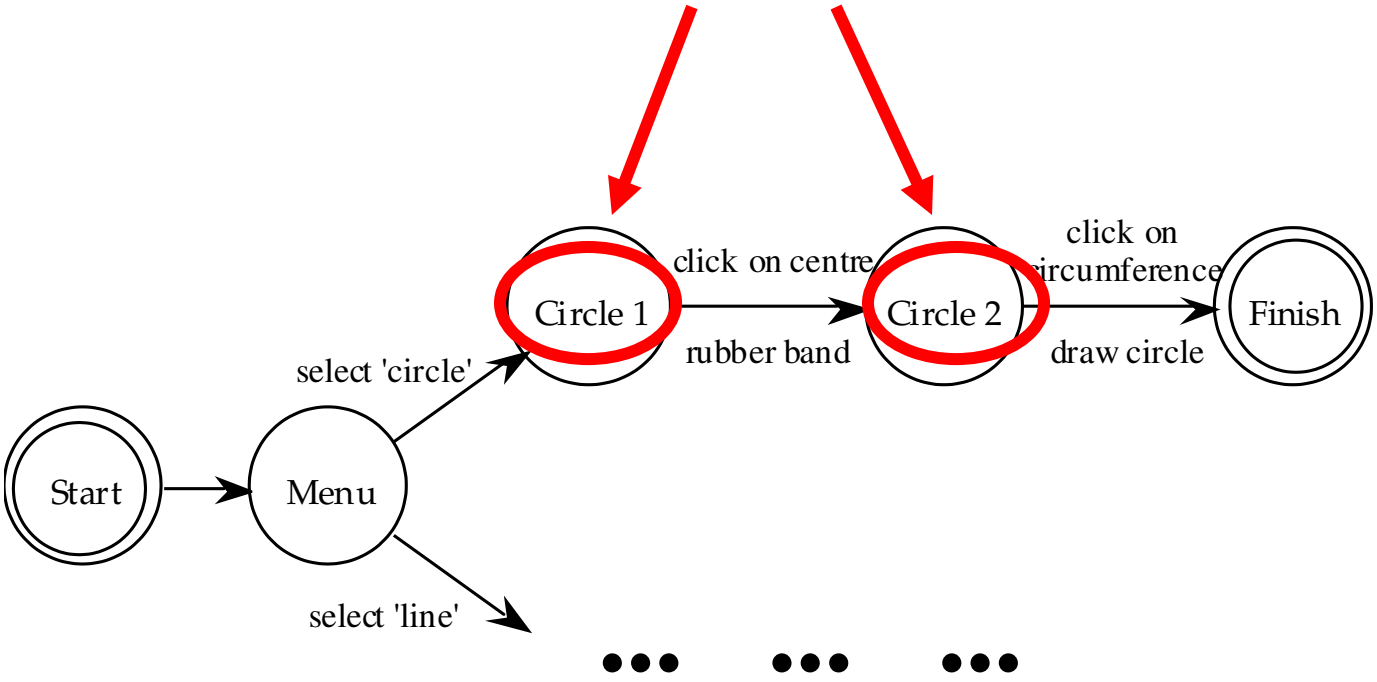
State transition networks - events

- arc labels a bit cramped because:
  - notation is `state heavy‘
  - the events require most detail



State transition networks - states

- labels in circles a bit uninformative:
  - states are hard to name
  - but easier to visualise



Modeling Dynamics – *Three-State Model*

MAXIM

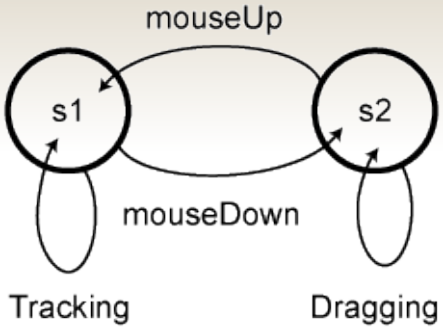
The Three-State Model can help designers to determine appropriate I/O devices for specific interaction designs

- The TSM can reveal intrinsic device states and their subsequent transitions
  - The interaction designer can use these to make determinations about the correlation between task and device
  - Certain devices can be ruled out early in the design process if they do not possess the appropriate states for the specified task

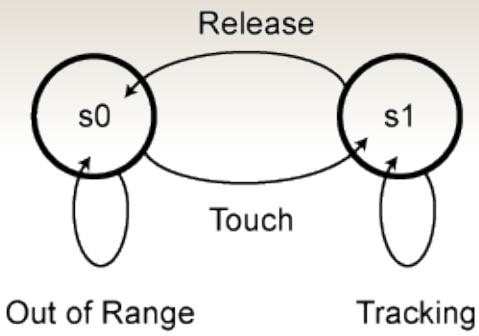
Modeling Dynamics – *Three-State Model*

- **The Three-State Model** (TSM) is capable of describing three different types of pointer movements
  - **Tracked**: A mouse device is tracked by the system and represented by the cursor position
  - **Dragged**: A mouse also can be used to manipulate screen elements using drag-and-drop operations
  - **Disengaged movement**: Some pointing devices can be moved without being tracked by the system, such as light pens or fingers on a touchscreen, and then reengage the system at random screen locations

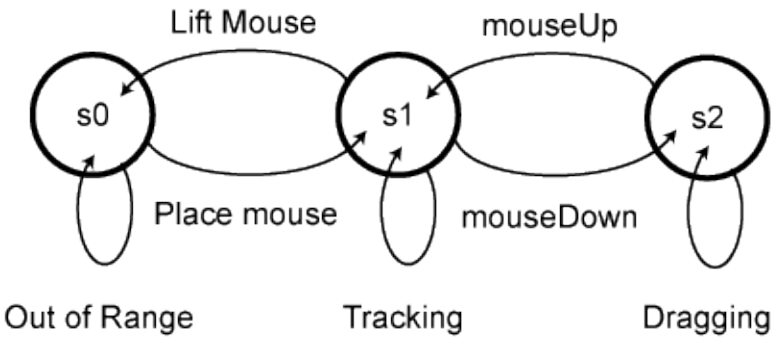
Modeling Dynamics – *Three-State Model*



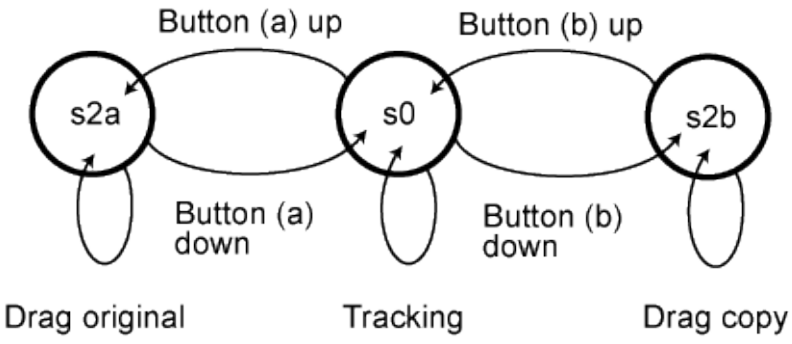
Mouse Three-State Model.



Trackpad Three-State Model.



Alternate mouse Three-State Model.

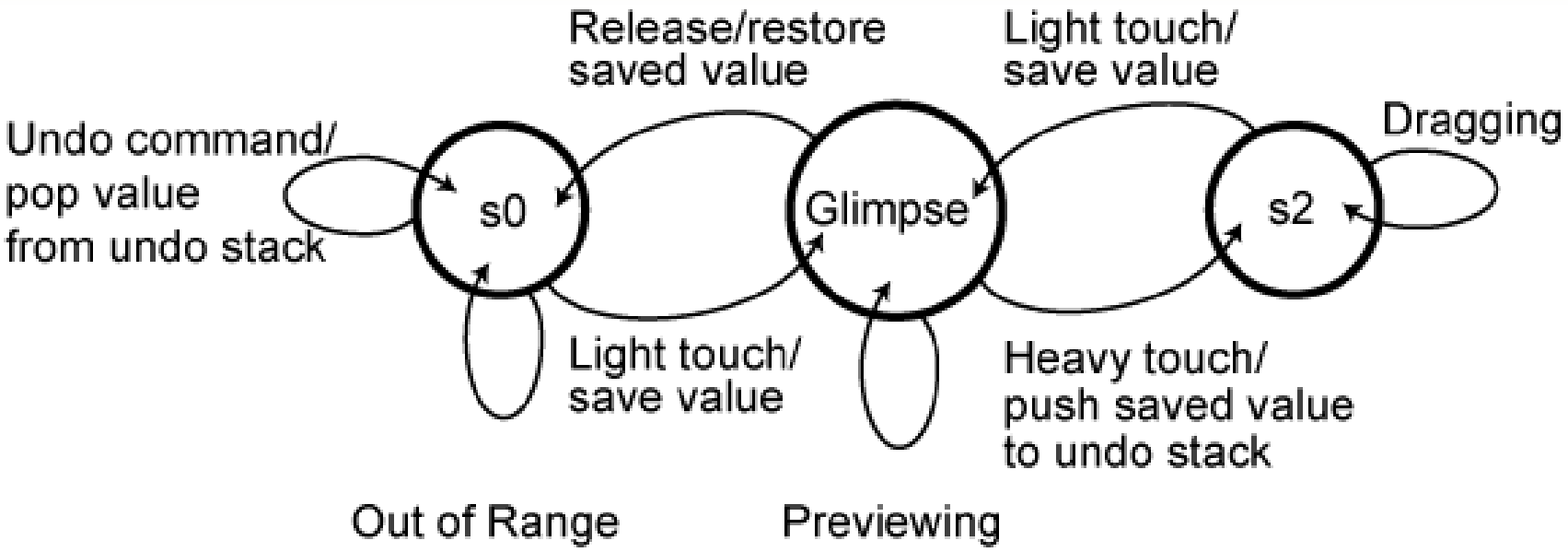


Multibutton pointing device Three-State Model.

Modeling Dynamics – *Glimpse Model*

- Forlines et al. (2005):
  - Because the pen and finger give clear feedback about their location when they touch the screen and enter state 2, it is redundant for the cursor to track this movement
  - Pressure-sensitive devices can take advantage of the s1 redundancy and map pressure to other features
  - Undo commands coupled with a preview function (Glimpse) can be mapped to a pressure-sensitive direct input device

Modeling Dynamics – *Glimpse Model*



Previewing potentially useful to scroll momentarily to another part of a document (but then return to where you were), or to look around in a virtual environment

Modeling Dynamics – *Glimpse Model*

- Some applications
  - **Pan and zoom interfaces**—Preview different magnification levels
  - **Navigation in a 3D world**—Quick inspection of an object from different perspectives
  - **Color selection in a paint program**—Preview the effects of color manipulation
  - **Volume control**—Preview different volume levels
  - **Window control**—Moving or resizing windows to view occluded objects
  - **Scrollbar manipulation**—Preview other sections of a document

Uses of State-Transition Networks

- Not well-suited to complete models of modern GUIs
  - Too many options (transitions) from any given state – combinatorial explosion (in fact, that's just the flexibility a good GUI is *supposed* to give)
- Better for limited/embedded user interfaces
  - Automated teller machine
  - Digital watch
  - Car key/alarm device
- Excellent for checking completeness of design
  - Be sure that all transitions are represented (and hence will get coded and tested in implementation)