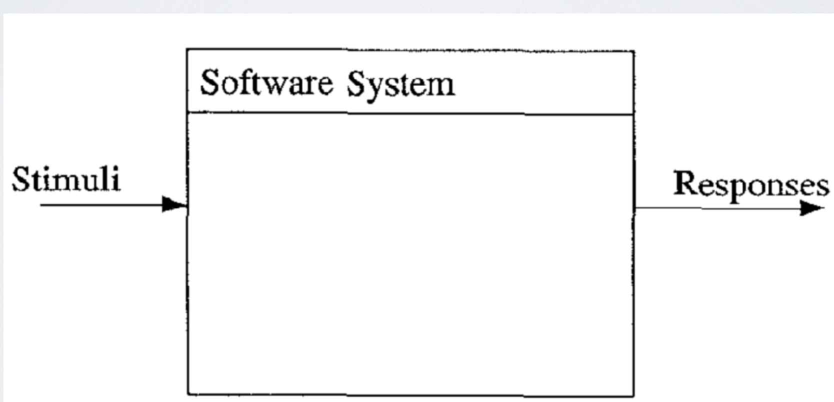
USAGE MODELS

- This discussion is based on the paper:
  - [Whi93] Whittaker James A., and J.H. Poore, ***Markov Analysis of Software Specifications***, ACM Transactions on Software Engineering and Methodology, Vol.2, No.1, January 1993, pp. 93-106.
  - We will discuss the paper for what it represents and later see how the approach can benefit us with respect to our "mission"
- The paper discusses Markov Chains as models for software usage
  - uses finite state discrete parameter Markov chain
  - states of the Markov chain represent entries from the input domain of the software
  - transitions (arcs) define ordering that determines the event space, or sequence, of the experiment

USAGE MODELS

- Black box view of software system

[Whi93, fig.1]



USAGE MODELS

- Markov analysis of software specifications
  - define underlying probability law for the usage of the software under consideration
  - analysis of specification done prior to design and coding
  - analysis yields irreducible Markov chain (usage Markov chain)
    - unique start state S_0
    - unique final state S_F
    - set of intermediate usage states S_i
    - states set $S = \{S_0, S_F\}$ union S_i
    - set S is ordered by probabilistic transition relation

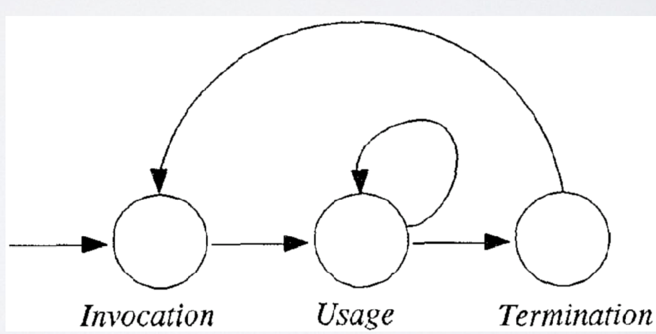
$$(S \times [0,1] \times S)$$

    - next state is independent of all past states given the present states
      - Markov property (first order chain)

USAGE MODELS

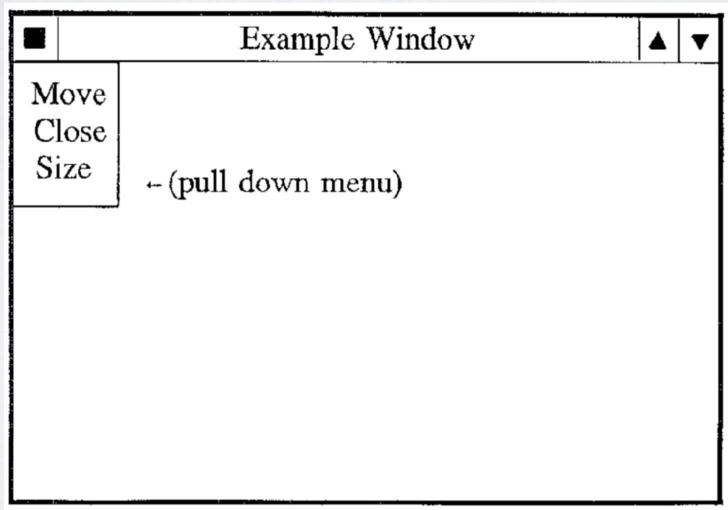
- Usage Markov chain has two properties
  - Structural Phase
    - the states and transitions of the chain are established
  - Statistical Phase
    - the transition probabilities are assigned

- Highest level transition diagram [Whi93, fig. 2]



USAGE MODELS

- Example: a simple window application [Whi93, fig3]



USAGE MODELS

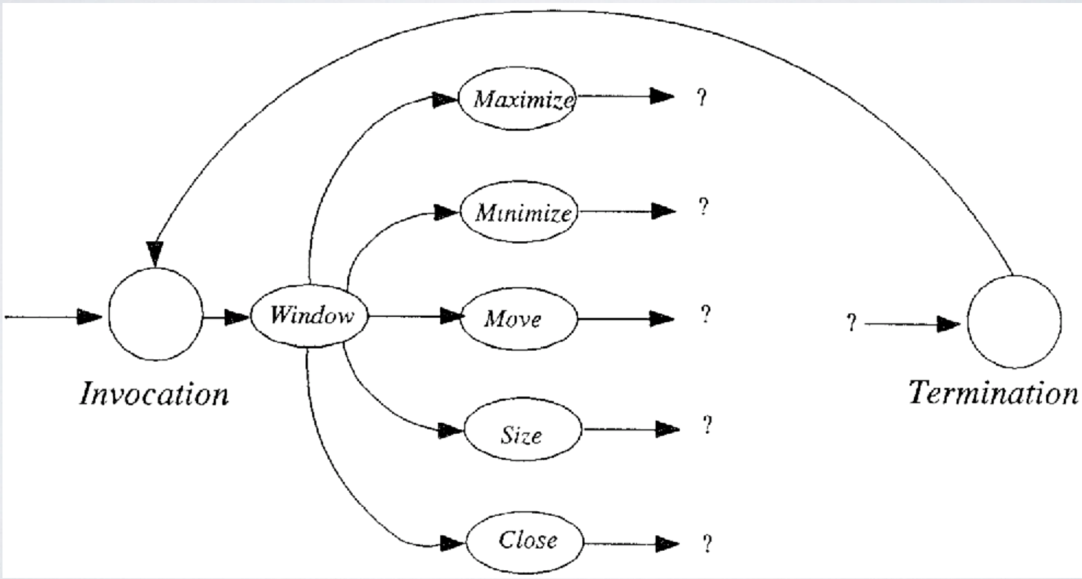
- Example Software Specification

[Whi93, table I]

Stimulus	Response
Invocation	Place the window of figure 3.2 on the screen
Select ▲	Expand the window dimensions to cover the entire area of the screen
Select ▼	Remove the window and replace it with its corresponding icon
Select ▪ and choose *Move* from the pull down menu	Move the window as directed by the mouse input (obeying screen boundaries)
Select ▪ and choose *Size* from the pull down menu	Size the window as directed by the mouse input (obeying minimum and maximum limits)
Select ▪ and choose *Close* from the pull down menu	Remove the window from the screen
Select the icon and release	Remove the icon from the screen and restore the window

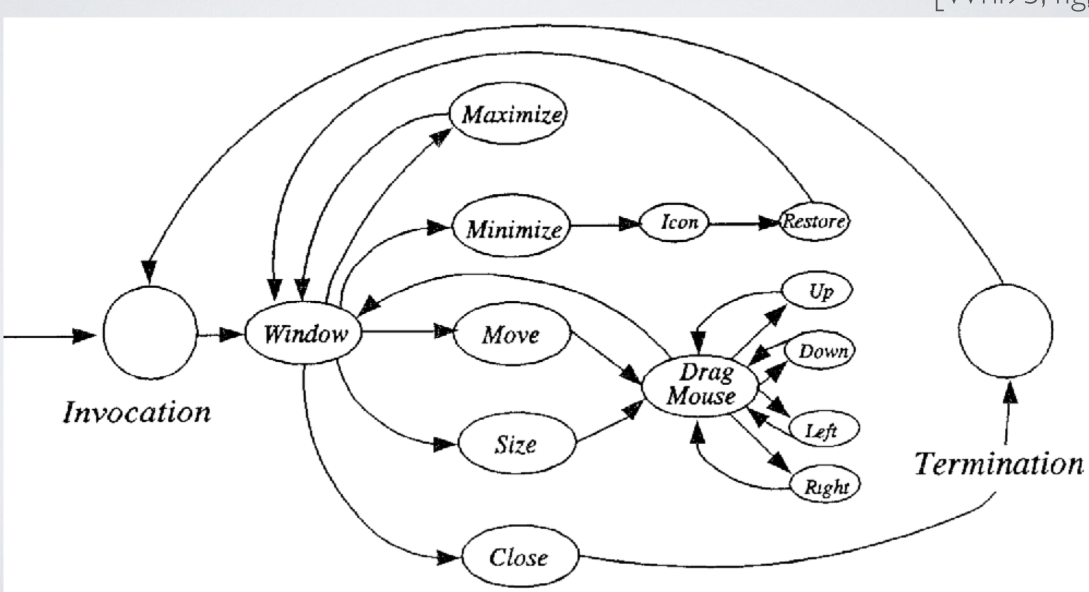
USAGE MODELS

- Expansion of the top level usage diagram [Whi93, fig, 4]



USAGE MODELS

- Structural phase - Constructing the usage Markov chain
  - phase is complete when usage is completely modeled [Whi93, fig,5]



USAGE MODELS

- **Statistical Phase**
  - assignment of transition probabilities
  - different approaches to statistical phase
    - **uninformed approach**
      - assign uniform probability distribution across the exit arcs for each state
      - useful when no information is available to make more informed choice

USAGE MODELS

- **Statistical Phase**
    - **informed approach**
      - when some actual user sequences are available
      - could be captured inputs from a prototype, or profiling information
      - resulting relative frequencies can be used to estimate the transition probability in the usage chain

USAGE MODELS

- **Statistical Phase**
  - **intended approach**
    - similar to informed approach but...
    - sequences are obtained by hypothesizing runs of the software by a careful and reasonable user
    - relative frequency estimates of transition probabilities are computed from the symbol transition counts as in the informed approach

- How does one rank the approaches?

USAGE MODELS

- Captured or hypothesized sequences [Whi93, table II]

1. *<Invocation><Window><Maximize><Window><Close><Termination>*
2. *<Invocation><Window><Minimize><Icon><Restore><Window><Close><Termination>*
3. *<Invocation><Window><Move><Drag Mouse><Down><Drag-Mouse><Right><Drag Mouse><Down><Drag Mouse><Window><Close><Termination>*
4. *<Invocation><Window><Size><Drag Mouse><Left><Drag-Mouse><Up><Drag Mouse><Left><Drag Mouse><Window><Close><Termination>*
5. *<Invocation><Window><Move><Drag Mouse><Down><Drag-Mouse><Left><Drag Mouse><Down><Drag Mouse><Window><Close><Termination>*
6. *<Invocation><Window><Size><Drag Mouse><Down><Drag-Mouse><Right><Drag Mouse><Window><Close><Termination>*

USAGE MODELS

- Assigning transition probabilities

From-State	To-State	Frequency	Probability
Invocation	*Window*	6	1
Window	*Maximize*	1	1/12
Window	*Minimize*	1	1/12
Window	*Move*	2	1/6
Window	*Size*	2	1/6
Window	*Close*	6	1/2
Maximize	*Window*	1	1
Minimize	*Icon*	1	1
Icon	*Restore*	1	1
Restore	*Window*	1	1
Move	*Drag Mouse*	2	1
Size	*Drag Mouse*	2	1
Drag Mouse	*Window*	4	4/15
Drag Mouse	*Up*	1	1/15
Drag Mouse	*Down*	5	1/3
Drag Mouse	*Left*	3	1/5
Drag Mouse	*Right*	2	2/15
Up	*Drag Mouse*	1	1
Down	*Drag Mouse*	5	1
Left	*Drag Mouse*	3	1
Right	*Drag Mouse*	2	1
Close	*Termination*	6	1
Termination	*Invocation*	-	1

[Whi93, table II]

(putting it on one page)

1. <*Invocation*><*Window*><*Maximize*><*Window*><*Close*><*Termination*>
2. <*Invocation*><*Window*><*Minimize*><*Icon*><*Restore*><*Window*><*Close*><*Termination*>
3. <*Invocation*><*Window*><*Move*><*Drag Mouse*><*Down*><*Drag-Mouse*><*Right*><*Drag Mouse*><*Down*><*Drag Mouse*><*Window*><*Close*><*Termination*>
4. <*Invocation*><*Window*><*Size*><*Drag Mouse*><*Left*><*Drag-Mouse*><*Up*><*Drag Mouse*><*Left*><*Drag Mouse*><*Window*><*Close*><*Termination*>
5. <*Invocation*><*Window*><*Move*><*Drag Mouse*><*Down*><*Drag-Mouse*><*Left*><*Drag Mouse*><*Down*><*Drag Mouse*><*Window*><*Close*><*Termination*>
6. <*Invocation*><*Window*><*Size*><*Drag Mouse*><*Down*><*Drag-Mouse*><*Right*><*Drag Mouse*><*Window*><*Close*><*Termination*>

From-State	To-State	Frequency	Probability
Invocation	*Window*	6	1
Window	*Maximize*	1	1/12
Window	*Minimize*	1	1/12
Window	*Move*	2	1/6
Window	*Size*	2	1/6
Window	*Close*	6	1/2
Maximize	*Window*	1	1
Minimize	*Icon*	1	1
Icon	*Restore*	1	1
Restore	*Window*	1	1
Move	*Drag Mouse*	2	1
Size	*Drag Mouse*	2	1
Drag Mouse	*Window*	4	4/15
Drag Mouse	*Up*	1	1/15
Drag Mouse	*Down*	5	1/3
Drag Mouse	*Left*	3	1/5
Drag Mouse	*Right*	2	2/15
Up	*Drag Mouse*	1	1
Down	*Drag Mouse*	5	1
Left	*Drag Mouse*	3	1
Right	*Drag Mouse*	2	1
Close	*Termination*	6	1
Termination	*Invocation*	-	1

USAGE MODELS

- Test Cases
  - Statistical Test Case
    - any connected state sequence of the usage chain begins in the start state and ends in the termination state
- Usage Distribution π
  - the structure of the usage chain induces a probability distribution on the input domain of the software
  - this distribution is called *usage distribution*
  - each state S_i has steady-state probability π_i
    - i.e., the probability of being in state *i* is π_i

USAGE MODELS

Usage Distribution π

- usage distribution can be computed by $\pi = \pi P$

- P is the transition matrix of the usage chain
  - P can be encoded as a 2-D matrix (P is a square matrix)
  - state labels are indices and transition probabilities are entries
  - each row sums up to one
  - each entry π_i is the expected appearance rate of state S_i in the long run
  - this tells software testers where the user spends most of its time
    - perhaps focus attention on these parts
    - there is a danger to this though, the bug may be in the less used functions
  - states can be grouped (allows comparison of subsections of software)
    - usage distributions are just summed up
    - collapsing states in a Markov chain may require adjustments to transitions

USAGE MODELS

- Other useful statistics
- Number of states necessary until S_i is expected to be generated, denoted by x_i, is computed by

$$x_i \pi_i = 1 \qquad \Rightarrow \qquad x_i = \frac{1}{\pi_i}$$

  - if S_i is the termination state, then x_i is the expected number of states until termination of the software

USAGE MODELS

- Expected number of sequences s_i necessary until state *i* occurs

$$s_i = \frac{x_i}{x_{TERM}} = \frac{\pi_{TERM}}{\pi_i}$$

  - largest element of vector *s* identifies the amount of expected testing until all usage states are encountered at least once
  - note: TERM indicates termination state

USAGE MODELS

[Whi93, table III]

- Analytical results for example usage model

State	π	x	s
Invocation	0 093750	10 7	1
Window	0.187500	5 3	0.5
Maximize	0 015625	64	6
Minimize	0 015625	64	6
Icon	0.015625	64	6
Restore	0.015625	64	6
Move	0 031250	32	3
Size	0.031250	32	3
Drag Mouse	0 234375	4.3	0 4
Up	0 015635	64	6
Down	0.078125	12 8	1.2
Left	0 046875	21.3	2
Right	0 031250	32	3
Close	0 093759	10.7	1
Termination	0 093750	10 7	1

USAGE MODELS

- **Mean first passage times m_{jk}**
  - m_{jk} is the expected number of usage states visited starting from S_j until the first visit to S_k

$$m_{jk} = 1 + \sum_{i \neq k} p_{ji} m_{ik}$$

  - p_{ij} indicate the transition probabilities
  - indicates the extent to which S_i and S_k are encountered within the same sequence
  - e.g. if m_{jk} is greater than the expected test case length, then
    - occurrence of S_j followed by S_k is expected to require multiple sequences
  - note: in figure of next slide the diagonal is vector *x*

USAGE MODELS

- Mean f

	Invocation	Window	Maximize	Minimize	Icon	Restore	Move	Size	Drag Mouse	Up	Down	Left	Right	Close	Term.
Invocation	11	1	64	62	63	64	26	26	11	74	22	31	42	9	10
Window	10	5	63	61	62	63	25	25	10	73	21	30	41	8	9
Maximize	11	1	64	62	63	64	26	26	11	74	22	31	42	9	10
Minimize	13	3	66	64	1	2	28	28	13	76	24	33	44	11	12
Icon	12	2	65	63	64	1	27	27	12	75	23	32	43	10	11
Restore	11	1	64	62	63	64	26	26	11	74	22	31	42	9	10
Move	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Size	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Drag Mouse	16	7	70	68	69	70	31	31	4	63	12	20	31	14	15
Up	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Down	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Left	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Right	17	8	71	69	70	71	32	32	1	64	13	21	32	15	16
Close	2	3	66	64	65	66	28	28	13	76	24	33	44	11	1
Termination	1	2	65	63	64	65	27	27	12	75	23	32	43	10	11

[Whi93, fig.6]

The mean first passage matrix for the example usage model (entries are rounded).

USAGE MODELS

- Source entropy of usage chain
  - the source entropy quantifies the uncertainty in a stochastic source
    - the entropy of a random variable *f* is the expected "surprise" of the event that *f(x)=y*

$$H = -\sum_i \pi_i \sum_j p_{ij} \log p_{ij}$$

  - again π is the usage distribution and p_{ij} is the transition probability

  - H is exponentially related to the number of sequences that are "statistically typical" of the Markov chain
    - a Markov chain has a set of typical sequences whose ensemble statistics closely match the statistics of the chain

USAGE MODELS

- Source entropy of usage chain
  - high H
    - => exponentially greater number of typical sequences
    - more sequences exist because of the uncertainty present in the model
    - => Markov chain must generate more sequences in order to accurately describe the Markov source

USAGE MODELS

- Source entropy of usage chain
  - source entropy serves as a comparative measure for chains with same structure but different probabilities
    - example: two chains U_1 and U_2 (chains are structurally the same)
      - transition probabilities of U_1 are ***uninformed***
      - transition probabilities of U_2 are ***informed***
      - Let H_1 and H_2 be the source entropies for U_1 and U_2 respectively
      - If $H_1 > H_2$ then one should expect exponentially greater number of sequences using U_1 than U_2
      - U_1 could serve as frame of reference
      - in previous example $H_1 = 1.0884$ and $H_2 = 0.8711$

USAGE MODELS

- Conclusions
  - Usage chains are a good tool trying to answer the question

    "What is the user likely to do when using the software?"

    or

    " What is the software to be able of doing?"
  - The paper was written to aid testing of software, not with survivability in mind
  - We need to determine how usage models can be used to benefit our "survivability" cause, e.g.,
    - How can we use usage models to define normal usage of the system?
    - How can we reverse-engineer usage patterns?
    - How can an attacker take advantage of usage models?