

Modeling the Disruption to the User's Mental Model



NIPS'06: User Adaptive Systems

Bowen Hui and Craig Boutilier
{bowen,cebly}@cs.utoronto.ca

Need for software customization

- increased software complexity

- customization examples
 - **functionality**: hide unused ones
 - **presentation**: ease of accessibility
 - **navigation**: thumbnails of slides

- most affected users
 - cognitive, sensory, motor impairments
 - elderly
 - children
 - novices

Adaptive interaction

- user learn system features, e.g.:
 - function location
 - process/wait time
 - execution procedure
- update knowledge into **mental model**
- **disruption** = adaptations contradicting MM
- consequences:
 - increased search time
 - user frustration
 - software dissatisfaction

SCA: Adaptive Software Customization

- methodology for online customization
 - predicts user goals
 - infers user personality/affect state
 - cost model
 - decision-theoretic action selection

- model complex tradeoffs:
 - partial help good for needy users
 - unfocused help may deter distractible users
 - frustrated users are not interruptible
 - independent users require little help

Individual cost model

- model tradeoffs in principled way
- quantification of rewards and costs
 - quantification of disruption
 - estimating user's mental model
- take decision-theoretic actions

Individual cost model

- model tradeoffs in principled way
- quantification of rewards and costs
 - quantification of disruption
 - estimating user's mental model
- take decision-theoretic actions

Mental models

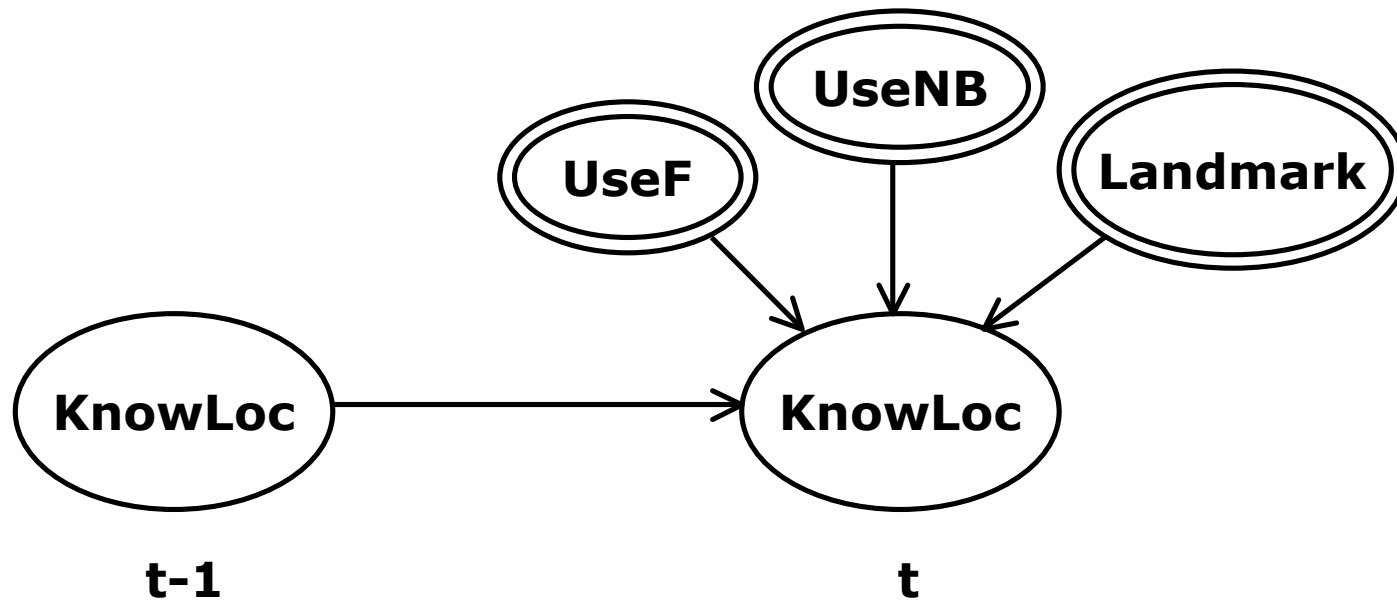
- characteristics (Norman 1983):
 - incomplete
 - dynamic
 - unstable

- existing computational models (Sasse 1997):
 - non-probabilistic
 - heuristic updates

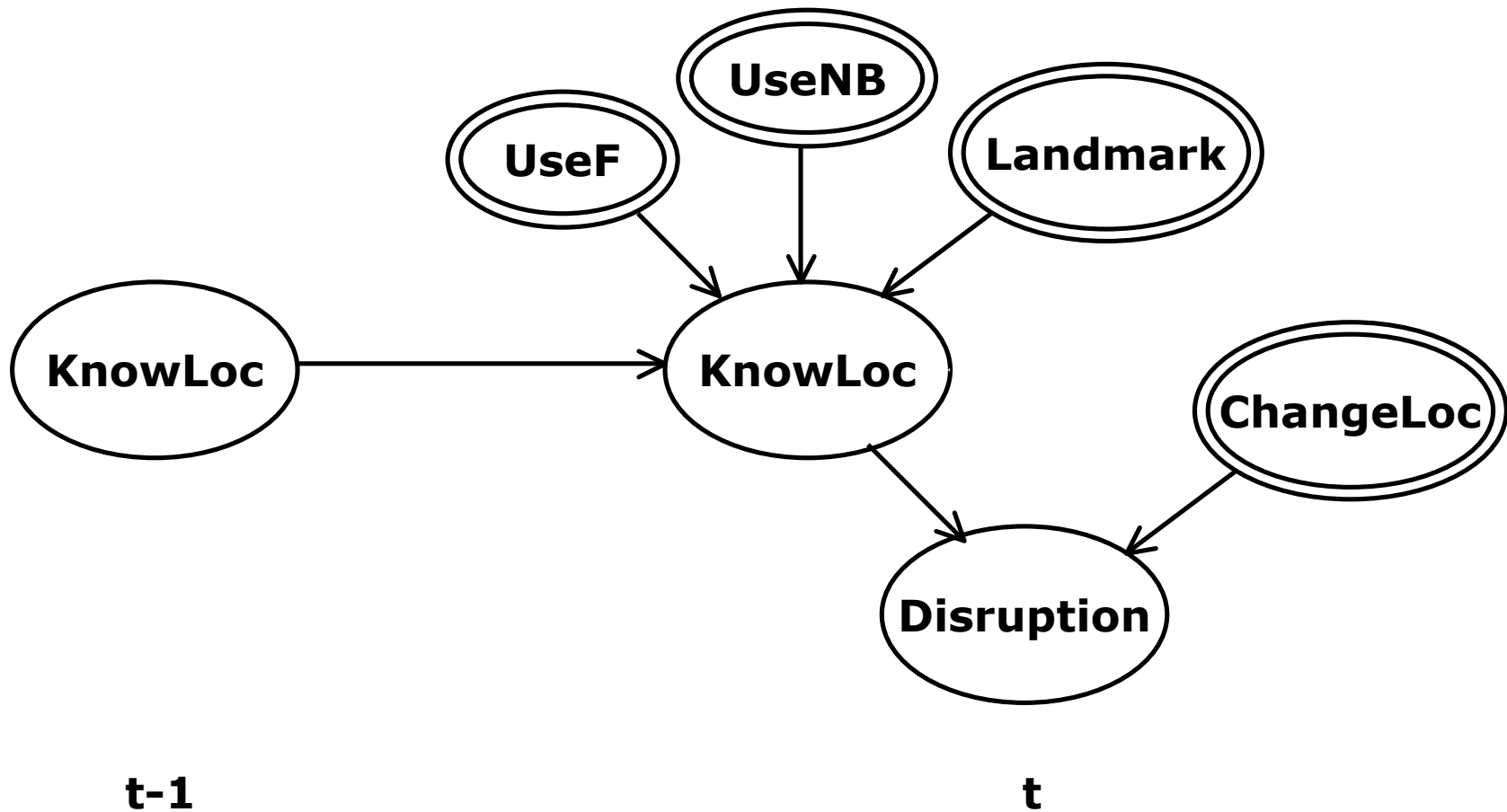
Probabilistic model

- use dynamic Bayesian network (DBN) to:
 - capture **unstable** knowledge
 - model **dynamic** updates
- extend DBN for disruption to:
 - model **incomplete** knowledge
- focus on adaptive function *locations*
- observations motivated by user experiments

DBN for mental model



DBN for disruption



User experiment

- PowerPoint menu structure (160 functions)
- 5 users



Task 1: training

- 10% functions into 4 frequency distributions

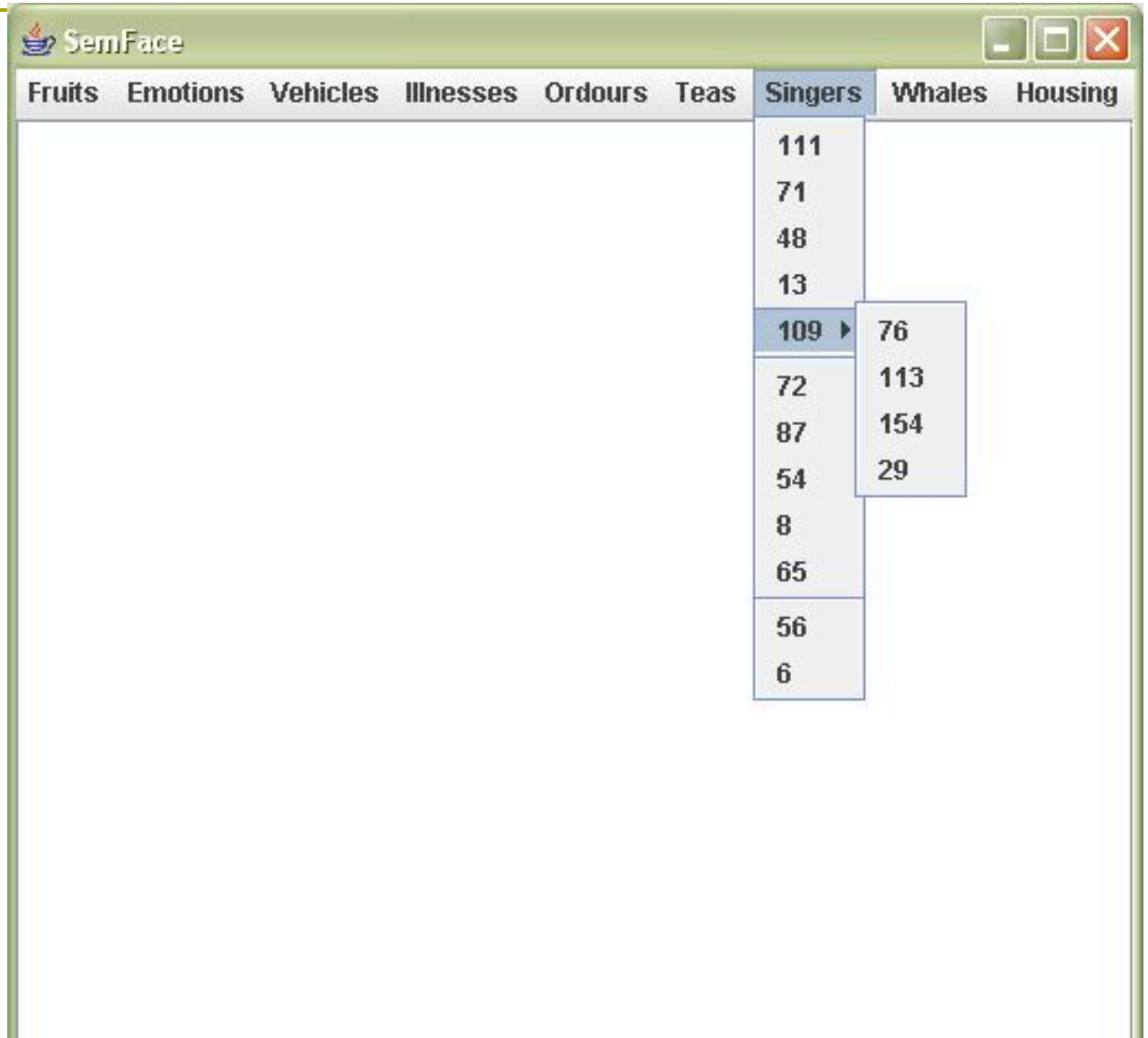


Task 1: results

- 16 functions
 - saw learning through time performance

Task 2: location recall

- 45 functions
 - 16 used
 - 15 NBs
 - 14 relatives



The screenshot shows a window titled "SemFace" with a menu bar containing "Fruits", "Emotions", "Vehicles", "Illnesses", "Ordours", "Teas", "Singers", "Whales", and "Housing". The "Singers" menu item is selected, and a dropdown list is visible. The dropdown list contains the following values: 111, 71, 48, 13, 109 (highlighted with a right-pointing arrow), 72, 87, 54, 8, 65, 56, and 6. A secondary dropdown menu is open for the value 109, showing the values 76, 113, 154, and 29.

Category	Value
Singers	111
Singers	71
Singers	48
Singers	13
Singers	109
Singers	72
Singers	87
Singers	54
Singers	8
Singers	65
Singers	56
Singers	6

Sub-category	Value
Sub-category of 109	76
Sub-category of 109	113
Sub-category of 109	154
Sub-category of 109	29

Tags used

- correct
- within region:
 - 1-up,1-down, other side
 - 2-up, 2-down, 3-up, 3-down
 - opposite-end (within same category)
- neighbouring category:
 - 1-category up/down, 2,3
- random:
 - top, middle, bottom
 - parent-menu, child-menu
- wrong menu

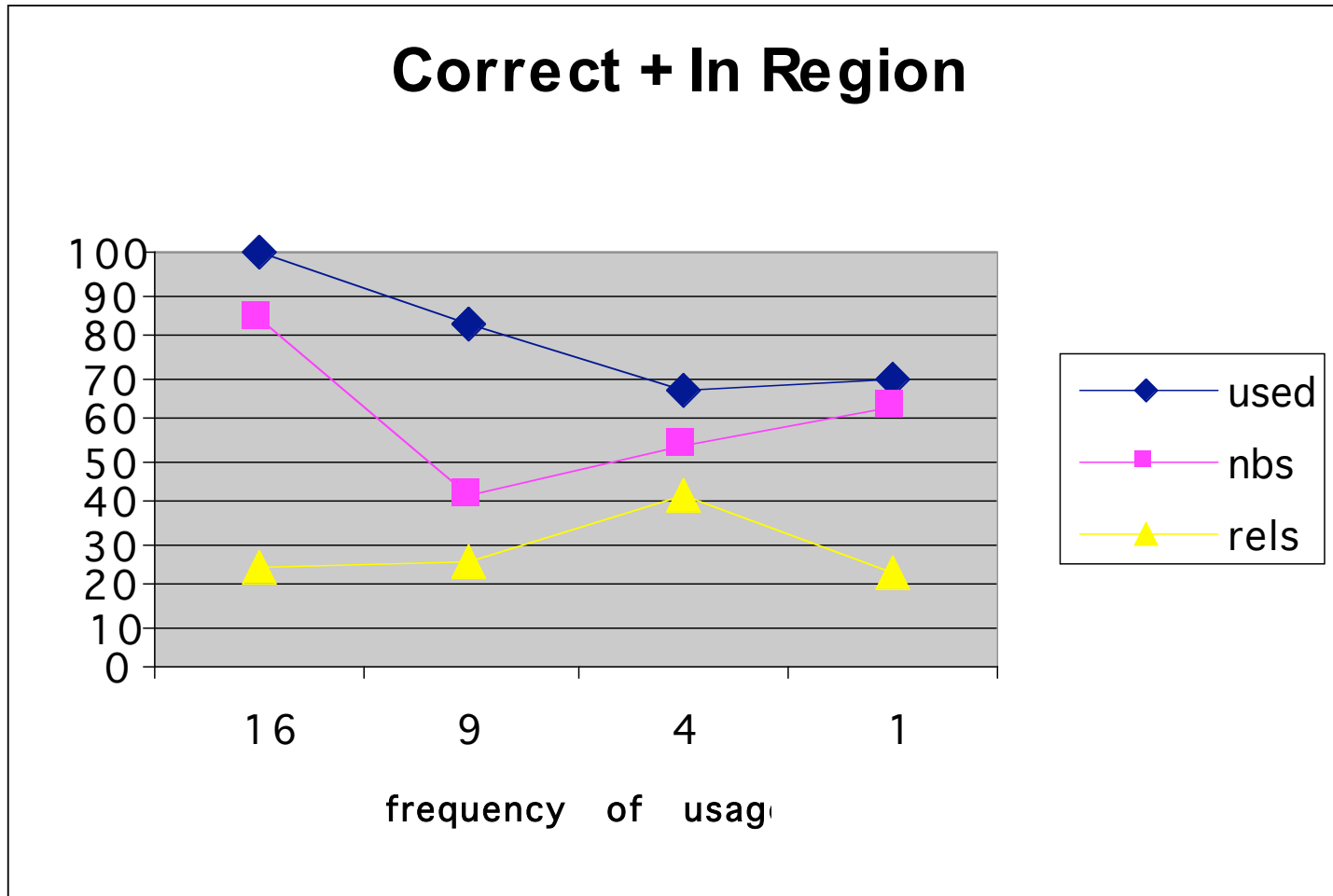
Region size vs. usage and frequency

	Used	Neighbours	Relatives
Frequency			
16	1	2	2
9	1+opp	2	2
4	2	3	2+opp
1	2+opp	3	3

Task 2: results

- location recall on 45 used and unused functions
 - frequency is strong key

Frequency vs. accuracy



Task 2: results

- location recall on 45 used and unused functions
 - frequency is strong key
 - some used landmarks

Landmark vs. accuracy

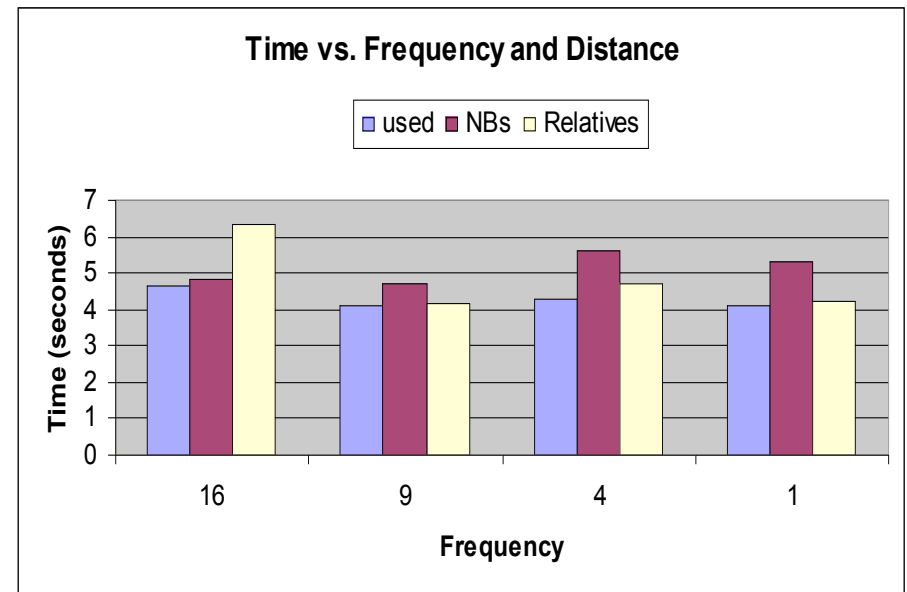
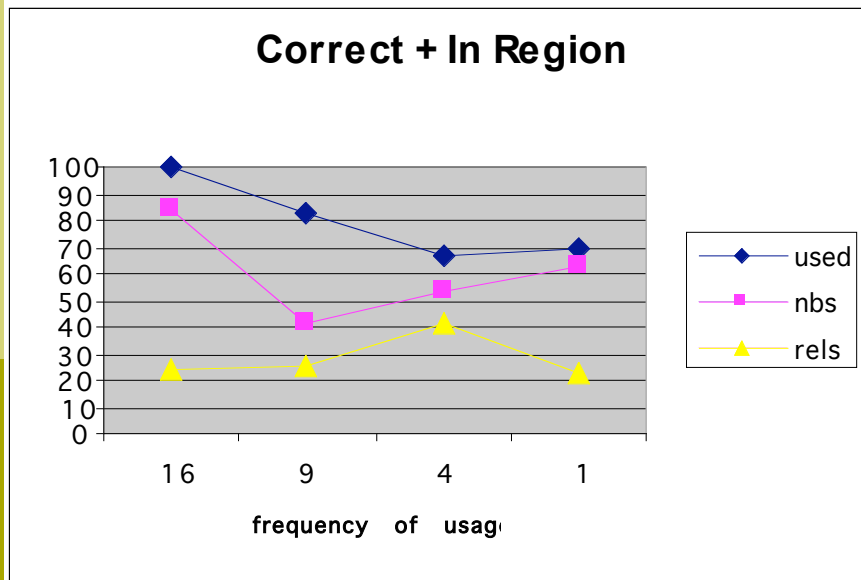
- exact matches over 65% accuracy:
 - first/last: u2,u3,u5
 - below >: u2,u3,u4
 - sub_first: everyone
- no effect:
 - above- (within region for u2,u3)
 - below-
 - sub_above-
 - sub_last

Task 2: results

- location recall on 45 used and unused functions
 - frequency is strong key
 - some used landmarks
 - some used neighbouring functions

Usage vs. accuracy and speed

- overall accuracy: used >> NBs >> relatives
- overall speed: used >> NBs



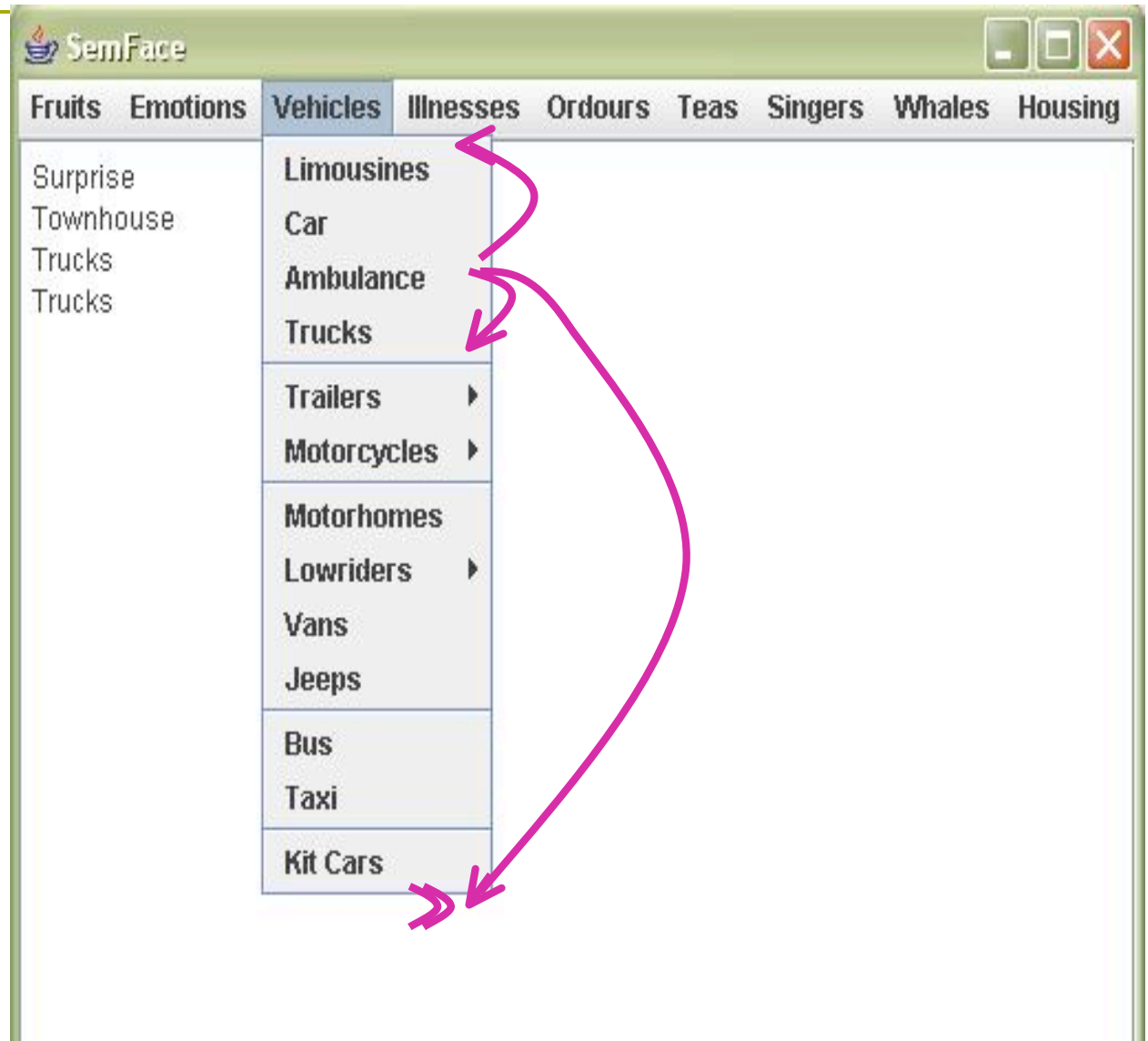
NBs helped

Task 2 results

- location recall on 45 used and unused functions
 - frequency is strong key
 - some used landmarks
 - some used neighbouring functions
- enriched DBN observations
- helped craft the priors used
- identified acceptable *region*

Task 3: eliciting disruption

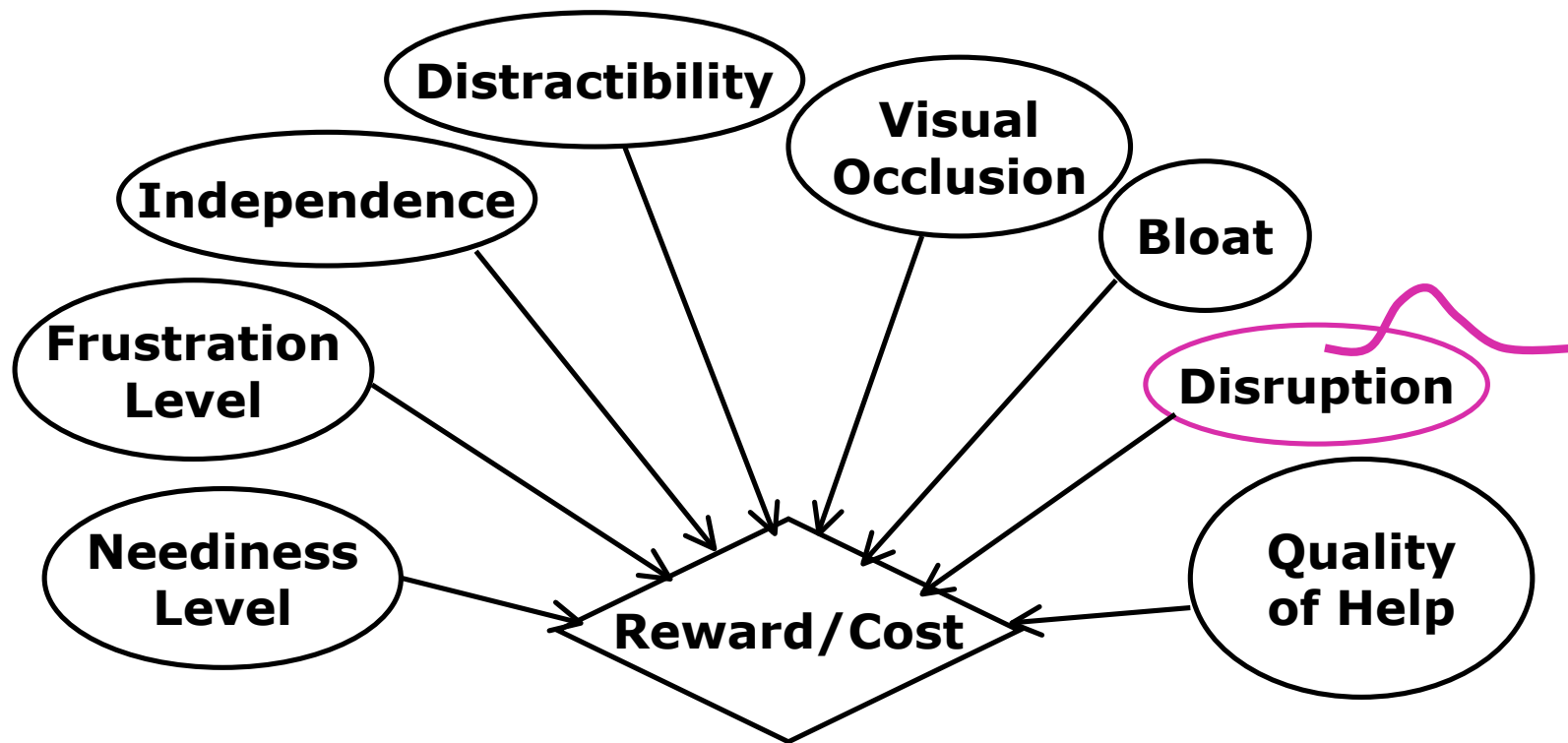
- 9 functions
 - 3 usages
 - 3 target locations
- Likert scale



Task 3 results

- eliciting disruption on 9 functions for 3 target locations
 - swapping neighbours is negligible/unnoticed
 - hiding “*unnecessarily*” is disruptive
 - results at top vary
(need to model convenience vs. disruption)
- used region to discretize ChangeLoc
- helped craft the priors used

SCA: Parametric cost model



Summary and future work

- model for disruption to mental model
- initial user experiments
 - support model structure
 - refine model details

- future work:
 - SCA framework as a POMDP
 - PowerPoint: implementation and test
 - still other user aspects to learn