Resilience of Interaction Techniques to Interruptions

A Formal Model-based Approach

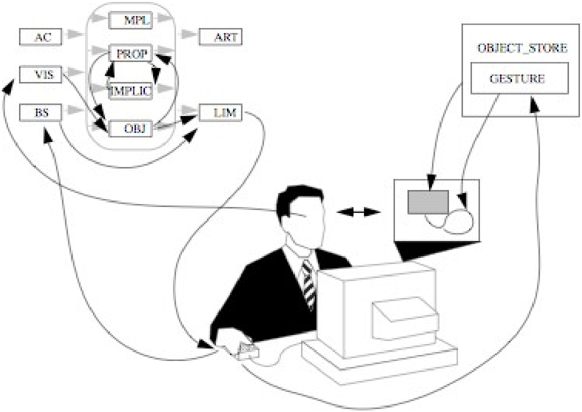
Mieke Massink

Consiglio Nazionale delle Ricerche
Ist. di Scienza e Tecnologie dell'Informazione "A. Faedo"
Formal Methods && Tools Lab

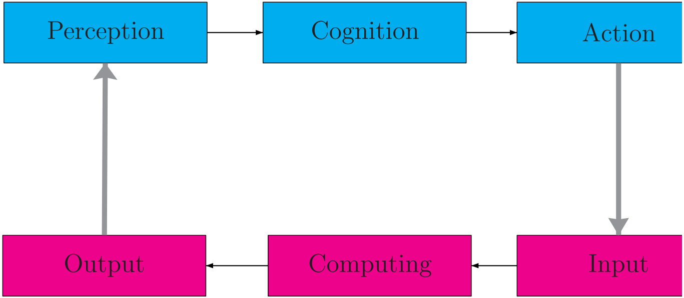
SEFM 2010 Course on Applications of FM in HCI (Part 2): Quantitative Models

2010

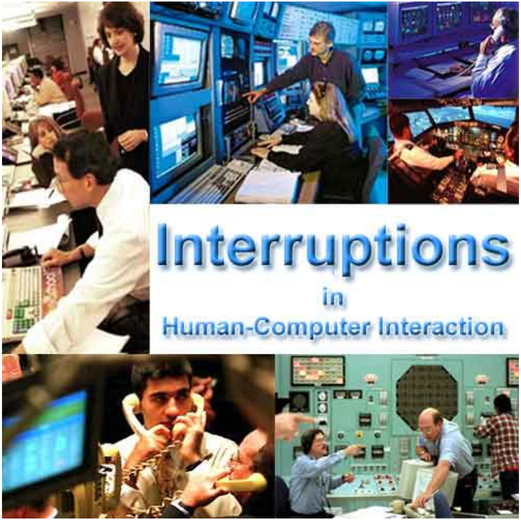
Human Computer Interaction



Human Computer Interaction



Introduction

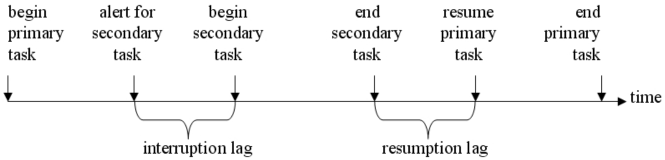


McFarlane and Latorella '02:
- Coordination
- Multi-tasking
- Dynamic Act.
- Cognition
- Design
- Taxonomy

Trafton et al. '03:
- Disruptiveness
- Anatomy
- Interruption lag
- LT memory

Bailey et al. '00:
- Workload
- Breakpoints

Anatomy of an interruption



Timeline: Anatomy of an interruption [Trafton03].

Resilient User Interfaces

Satellite Uplink Control Center:
Removing Objects from current System Presentation using Multi-modal man-machine interfaces

Longer Term Objective

Our research question:

Are formal stochastic models suitable to analyse and predict the resilience of multi-modal interfaces to interruptions?

Main stream:
empirical studies

- Real observations
- Costly to organise
- Time consuming
- Often prohibitive in design phase

Formal methods:
Models:

- Exploration of many situations
- Re-use of empirical results for ‘sub-tasks’
- Approximative
- Predictiveness

Outline

1. Introduction

Outline

1. Introduction
2. Case Study: Drag ’n Drop, Speak ’n Drop

Outline

1. Introduction
2. Case Study: Drag ’n Drop, Speak ’n Drop
3. Methodology

Outline

1. Introduction
2. Case Study: Drag ’n Drop, Speak ’n Drop
3. Methodology
4. Stochastic User models

Outline

Outline

Outline

Resilience to interruptions

Case study:

1. Primary task: Icon removal from desktop
2. Secondary: deal with interruptions by pop-up windows
3. Two interaction techniques: Drag'n'Drop and Speak'n'Drop
4. S'n'D is Multi-modal: Mouse and Voice

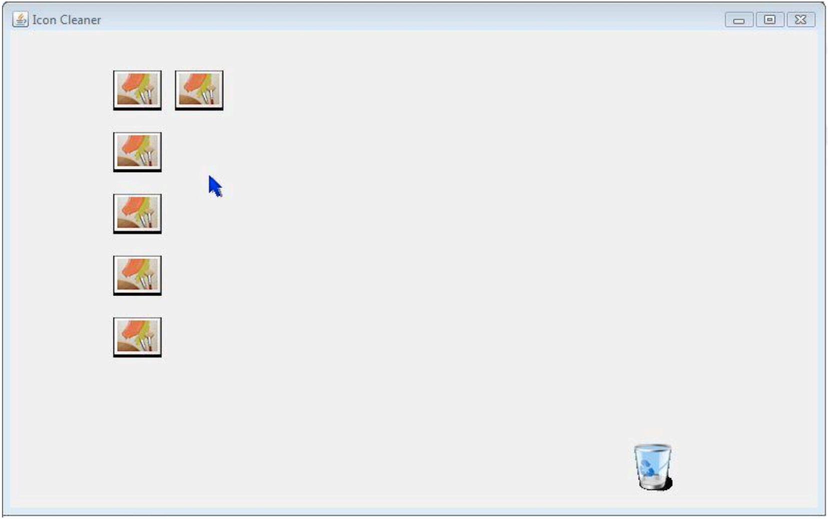
Goal:

1. (Stochastic) Model based analysis of resilience to interruptions
2. Comparing average number of successful removals for both interaction techniques by varying interruption rates
3. Viability and Predictive capability of Stochastic Models

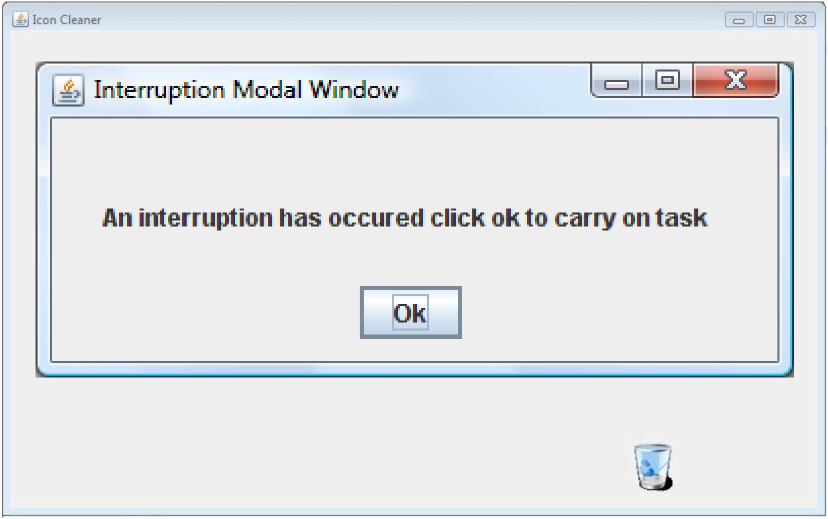
Approach

1. Stochastic Model of user part informed by
   - cognitive theory (ICS, Barnard 1985) and
   - human factors (e.g. Fitts' Law studies)
2. Joint stochastic model comprising behaviour of:
   - user
   - system and
   - interruptions
3. Language: Stochastic Automata (Stoch. Process Algebra PEPA)
4. Tool: Stochastic Model Checker PRISM
5. Analysis:
   - Comparison of number of drops for different interrupt rates
   - Validation of approach: comparison with empirical data

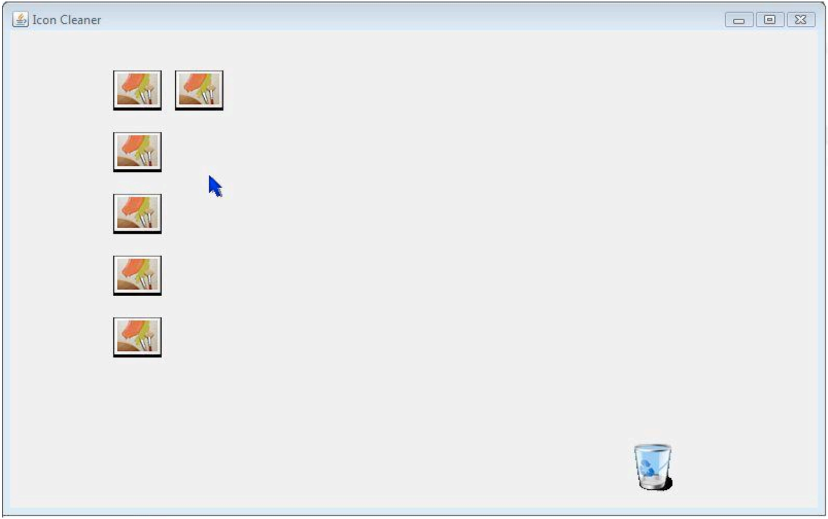
Case study: removing items from desktop



Case study: removing icons from desktop

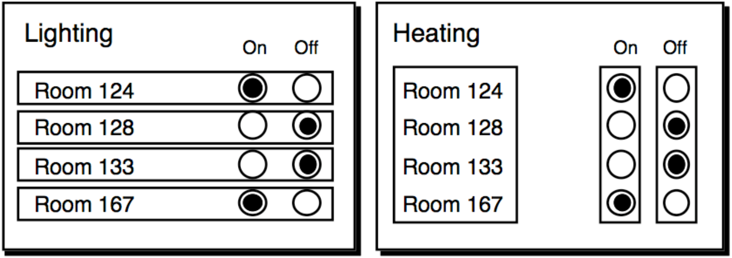


Case study: removing items from desktop

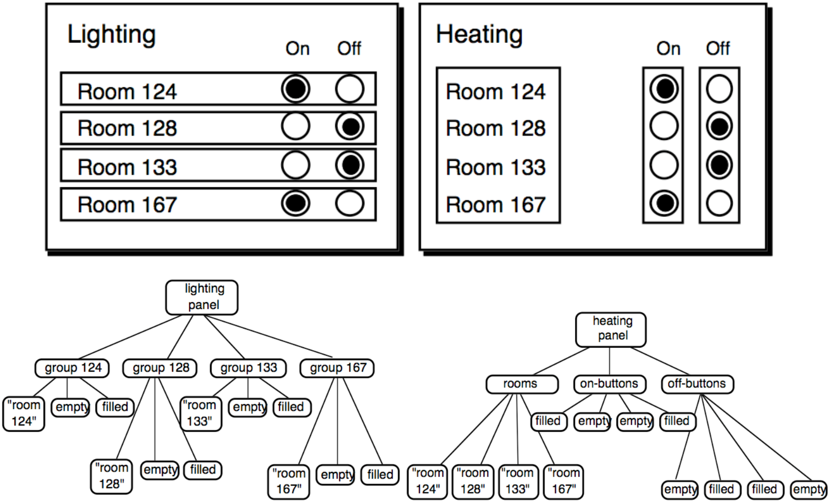


(Cognitive) Activities involved in DnD

- Visual observation of desktop
- Selection of icon to remove
  - Planning
  - Ballistic
  - Approach (visual control)
  - Adjust (optional, visual control)
- Drag to trash (also 3-4 phases)
- Release button

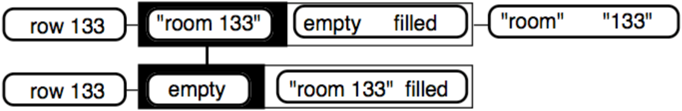

Lighting
On
Off
Room 124
Room 128
Room 133
Room 167
Heating
On
Off
Room 124
Room 128
Room 133
Room 167

Cognition: How we perceive, think and act



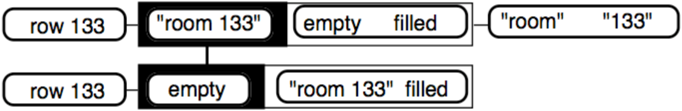
Cognition: How we perceive, think and act

Light in room 133:

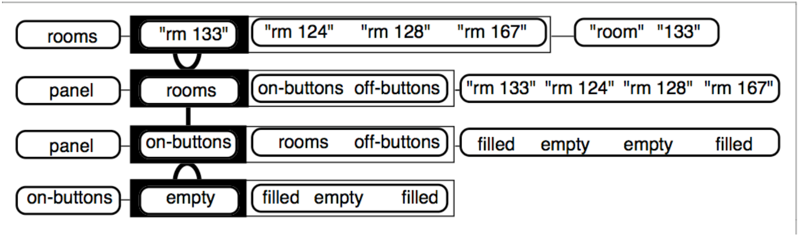


Cognition: How we perceive, think and act

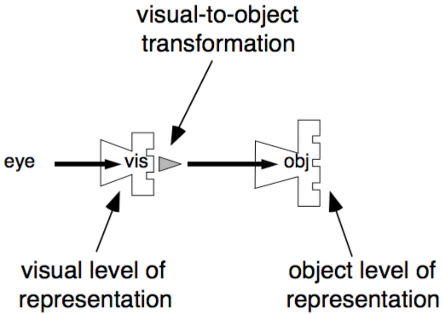
Light in room 133:



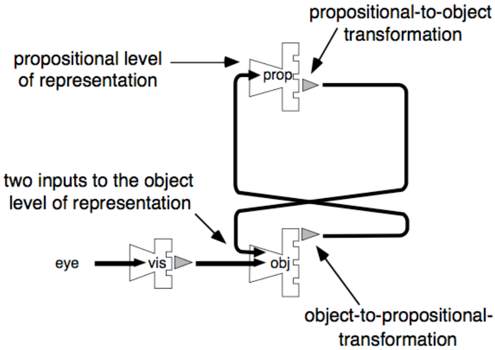
Heat in room 133:



Cognition: How we perceive, think and act



Cognition: How we perceive, think and act



Cognition: How we perceive, think and act

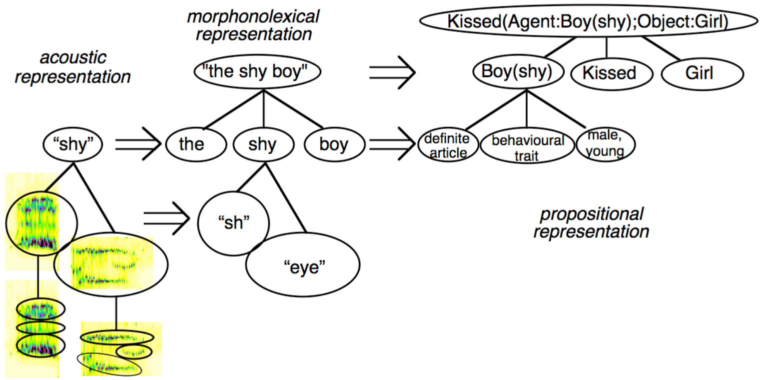
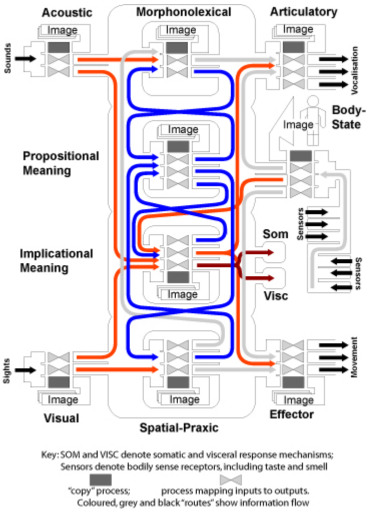


Figure 7.1 : the transformations required to understand speech produce successively more abstract representations by moving up the structure

Interacting Cognitive Subsystems [Barnard & May, 1993]



- Focus of attention
- Proceduralisation
- Blending
- Parallel processing
- Feedback loops

Formalisation of ICS theory

interactor ICS
attributes

$$
\begin{array}{lcl}
\textit{sources} & : & \textit{tr} \longrightarrow \mathbb{P}\,\textit{tr} \\
\textit{stable} & : & \mathbb{P}\,\textit{tr} \\
\_@\_ & : & \textit{code} \leftrightarrow \textit{sys} \\
\textit{coherent} & : & \mathbb{P}\,\mathbb{P}\,\textit{tr} \\
\textit{buffered} & : & \textit{tr} \\
\textit{config} & : & \mathbb{P}\,\textit{tr}
\end{array}
$$

actions

$$
\begin{array}{lcl}
\textit{engage} & : & \textit{tr} \times \textit{tr} \\
\textit{disengage} & : & \textit{tr} \times \textit{tr} \\
\textit{buffer} & : & \textit{tr} \\
\textit{trans} & &
\end{array}
$$

Formalisation of ICS

axioms

1 $\forall\, trs : \mathbb{P}\, tr \bullet trs \in coherent$

$\Leftrightarrow$

$$\exists\, dest : sys \bullet \left(\begin{array}{l} \forall\, s, t : sys \bullet \text{:s-t:} \in trs \Rightarrow t = dest \\ \wedge \\ \forall\, s, t : sys;\ p, q : repr \bullet \left(\begin{array}{l} \text{:s-dest:} \in trs \wedge p_s@dest \\ \wedge \\ \text{:t-dest:} \in trs \wedge q_t@dest \end{array} \right) \Rightarrow p \approx q \end{array} \right)$$

2 $t \in stable \Leftrightarrow sources(t) \in coherent \wedge (t = buffered \vee sources(t) \subseteq stable)$

3 $t \in config \Leftrightarrow (t \in stable \wedge \exists\, s \bullet t \in sources(s))$

4 $\mathbf{per}(engage(t, src)) \Rightarrow src \in stable$

5 $$t \notin stable \Rightarrow \left(\begin{array}{l} \exists\, s \bullet s \in stable \wedge s \notin sources(t) \wedge \mathbf{obl}(engage(t, s)) \\ \vee \\ \exists\, s \bullet s \notin stable \wedge s \in sources(t) \wedge \mathbf{obl}(disengage(t, s)) \\ \vee \\ \mathbf{obl}(buffer(t)) \end{array} \right)$$

6 $[buffer(t)]\ buffered = t$

7 $sources(t) = S \Rightarrow [engage(t, s)]\ sources(t) = S \cup \{s\}$

8 $sources(t) = S \Rightarrow [disengage(t, s)]\ sources(t) = S - \{s\}$

9 $p_x@src \wedge \text{:src-dst:} \in stable \Rightarrow [trans]\ p_{src}@dst$

10 $(\exists\, p : repr;\ src, dst : sys \bullet [trans]\ p_{src}@dst) \Rightarrow \exists\, x : sys \bullet p_x@src \wedge \text{:src-dst:} \in stable$

11 $\forall\, s, t : sys \bullet \text{:s-t:}, \text{:t-s:} \in config \wedge (buffered = \text{:s-t:} \vee buffered = \text{:t-s:}) \wedge x : tr \bullet x \notin stable$
$\Rightarrow \mathbf{per}(buffer(x))$

Formalisation of ICS: Axiom 1

axioms

1 $\forall\, trs : \mathbb{P}\, tr \bullet trs \in coherent$

$\Leftrightarrow$

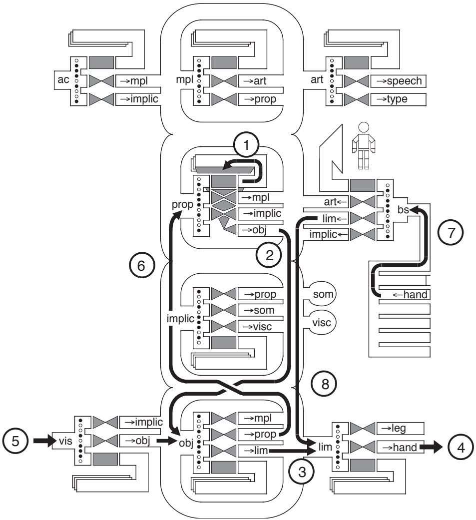
$$\exists\, dest : sys \bullet \left(\begin{array}{l} \forall\, s, t : sys \bullet \text{:s-t:} \in trs \Rightarrow t = dest \\ \wedge \\ \forall\, s, t : sys;\; p, q : repr \bullet \left(\begin{array}{l} \text{:s-dest:} \in trs \wedge p_s @ dest \\ \wedge \\ \text{:t-dest:} \in trs \wedge q_t @ dest \end{array} \right) \Rightarrow p \approx q \end{array} \right)$$

A set of transformation processes *trs* are coherent iff

1) they all produce data for the same destination subsystem, and

2) all the representations they produce can be blended

ICS configuration visual control



Mental **configuration** to operate a mouse (visual control):

1. Goal refresh
2. Object code
3. Limb code
4. Activate hand
5. Visual input
6. Propositional
7. Proprioceptive
8. Hand feedback

40 ms. per transformation

Fitts' Law

Movement time (MT) depends on **D**istance and **W**idth of object:

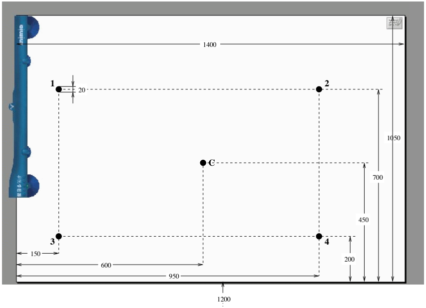
$$MT = a + b \log_2 \left(\frac{D}{W} + 1 \right)$$

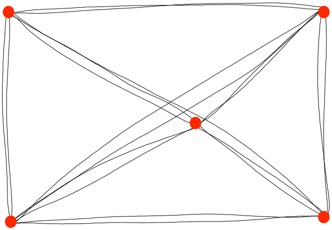
where
a is the start/stop time of the device
b is the inherent speed of the device

Movement has different phases [Faconti & Massink, 2007]:

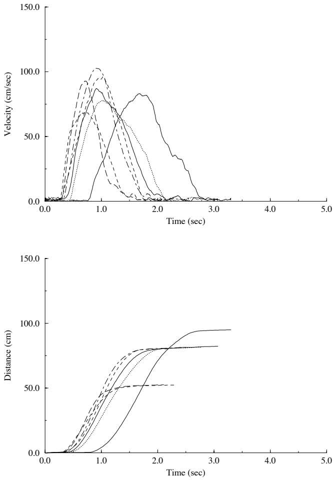
- planning
- ballistic
- approaching (under visual control)
- adjustment (under visual control, optional)

Movement time study [Faconti&Massink, 2007]

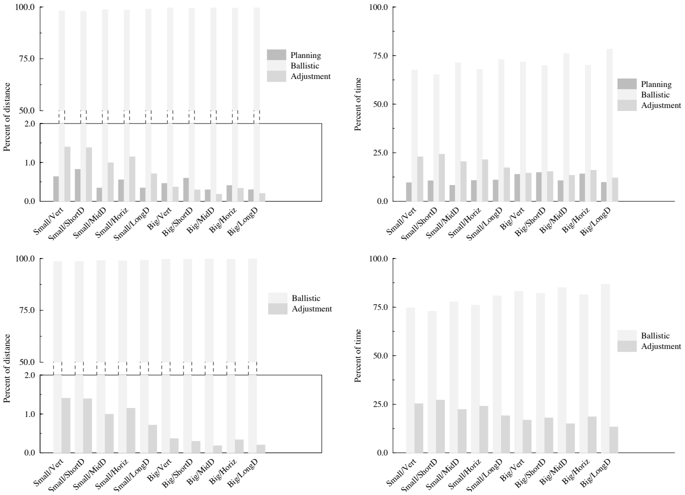




Movement time study (2)



Movement time study (3)



Movement time study (4)

On a 17 inch monitor:

- 968 ms on average to point to any position
- ballistic phase: 69% of time
- approach and adjustment phase: 31% of time

Only mean time is known: parameter of exponential distribution

$$Prob(s, t) = 1 - e^{-\lambda * t}$$

where $1/\lambda$ represents the mean time

Performance Evaluation PA - PEPA [Hillston '96]

A Markovian extension of a subset of CSP

- Prefix:$(\alpha, \lambda).P$; the duration of activity of type α is a random variable exponentially distributed with *rate* λ
- Choice: $P + Q$; models choice based on the *race condition* principle;
- Cooperation:$P \bowtie_L Q$; CSP-like multiparty synchronisation; the rate of a shared activity (in L) is the *min* of the (apparent) rates of the cooperating activities in the components:

$$A = ((\alpha, r1).0 + (\alpha, r2).0) \bowtie_{\alpha} (\alpha, r3).0$$

The apparent rate of α in $(\alpha, r1).0 + (\alpha, r2).0$ is $r1 + r2$
The rate of α in A is $min(r1 + r2, r3)$.

- passive actions: The rate of an action can also be left unspecified on one side of the cooperation:

$$B = (\beta, \top).0 \bowtie_{\beta} (\beta, r1).0$$

the total rate of β in B is than $min(r1, \top) = r1$. Note that the following holds:

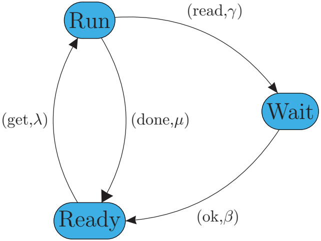
$$\begin{array}{lcll} m\top & < & n\top & \text{if } m < n \text{ and } m, n \in \mathbb{Q} \\ r & < & n\top & \text{with } r \in \mathbb{R} \text{ and } n \in \mathbb{Q} \\ m\top + n\top & = & (m+n)\top & \text{m,n} \in \mathbb{Q} \\ \frac{m\top}{n\top} & = & \frac{m}{n} & \text{m,n} \in \mathbb{Q} \end{array}$$

- Constants: $(A = P)$ for (recursive) process definitions and
- Hiding: (P/L) turning into τ any action in L.

Stochastic Model Checking: Models

Based on Labelled Continuous Time Markov Chains

- $(\mathcal{S}, \mathbf{R}, \mathbf{L})$ where
  - $\mathcal{S}$ finite set of states and
  - $\mathbf{R} : \mathcal{S} \times \mathcal{S} \to \mathrm{R}_{\geq 0}$ is the *rate matrix*
  - $\mathbf{L} : \mathcal{S} \to 2^{AP}$ is the labelling function
- Rate is parameter of Exponential distribution of random variable X denoting the probability that it takes at most t time units to move from s to s'
- if $\mathbf{R}(s, s') \neq 0$ then there is a transition $s \to s'$ and $\mathrm{Prob}(X \leq t) = 1 - e^{-\mathbf{R}(s,s')\cdot t}$
- Exponentially distributed random variables are 'memoryless'
- We use CTMC's that allow 'selfloops'
- CTMC's are generated from high level specifications, e.g. PEPA
- Stochastic model checkers: e.g. PRISM and ETMCC



$$Q = \begin{pmatrix} -\lambda & \lambda & 0 \\ \mu & -(\mu+\gamma) & \gamma \\ \beta & 0 & -\beta \end{pmatrix}$$

Ready	=	(get,λ).Run
Run	=	(done,μ).Ready + (read,γ).Wait
Wait	=	(ok,β).Ready

Stochastic Model Checking: Logic

Continuous Stochastic Logic (CSL):

State formulae: $\Phi ::= a \mid \neg\Phi \mid \Phi \vee \Phi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\varphi)$

$\mathcal{S}_{\bowtie p}(\Phi)$: probability that Φ holds in steady state $\bowtie p$
$\mathcal{P}_{\bowtie p}(\varphi)$: probability that a path fulfills $\varphi \bowtie p$

Path formulae: $\varphi ::= X^I\,\Phi \mid \Phi\,\mathcal{U}^I\,\Phi$

$X^I\,\Phi$: next state is reached at time $t \in I$ and fulfills Φ
$\Phi\,\mathcal{U}^I\,\Psi$: Φ holds along path until Ψ holds at $t \in I$

Examples:

$$\mathcal{P}_{\geq 0.5}((\neg Wait)\,\mathcal{U}\,Ready)$$
$$\mathcal{P}_{\leq 0.1}((Run \vee Ready)\,U^{[10,20]}\,Wait)$$
$$\mathcal{S}_{\geq 0.8}(Run)$$

PEPA Model Checking

stochastic system

quantitative requirements

stochastic model $\mathcal{M}$

formula Φ of some stochastic logic

stochastic model checker

quantitative analysis of $\mathcal{M}$ against Φ

probability for "bad behaviors" is $< \mathbf{10^{-6}}$

probability for "good behaviors" is $\mathbf{1}$

expected costs for ....

6 / 7

Continuous Stochastic Reward Logic

CSL with rewards:

$R\,\text{bound}\,[\,\text{rewardprop}\,]$

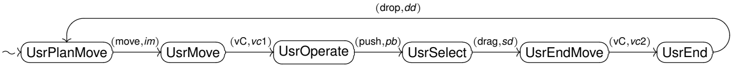
PRISM has four different types of reward properties:

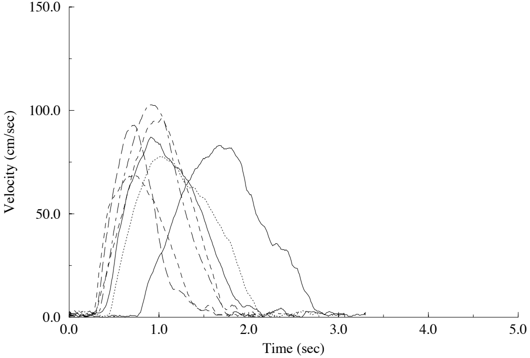
- reachability reward: F prop
- cumulative reward: $C \leq t$
- instantaneous reward: $I = t$
- steady state reward: S

Examples:

- $R_{\leq 9.5}\,[F\,z = 2\,]$
- $R_{=?}\,[C \leq 300\,]$
- $R_{\leq 4.4}\,[I = 100\,]$
- $R_{\leq 0.7}\,[S]$

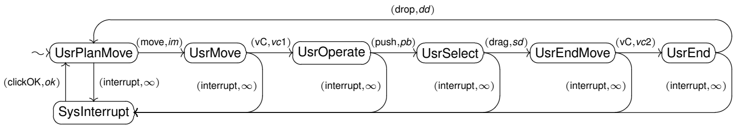
Drag'n'drop model: User



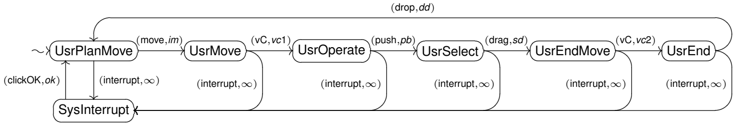


Transitions model time to change cognitive configuration

Drag’n’drop model: User



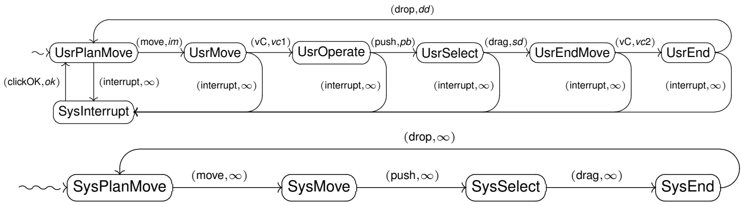
Drag and drop model in PEPA: User



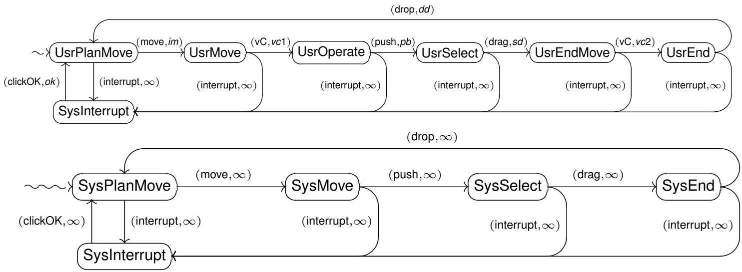
In PEPA:

UsrPlanMove = (move,im).UsrMove + (interrupt,infty).UsrInterrupt;
UsrMove = (visualControl,vc1).UsrOperate + (interrupt,infty).UsrInterrupt;
UsrOperate = (push,pb).UsrSelect + (interrupt,infty).UsrInterrupt;
UsrSelect = (drag,sd).UsrEndMove + (interrupt,infty).UsrInterrupt;
UsrEndMove = (visualControl,vc2).UsrEnd + (interrupt,infty).UsrInterrupt;
UsrEnd = (drop,dd).UsrPlanMove + (interrupt,infty).UsrInterrupt;
UsrInterrupt = (clickOK,infty).UsrPlanMove;

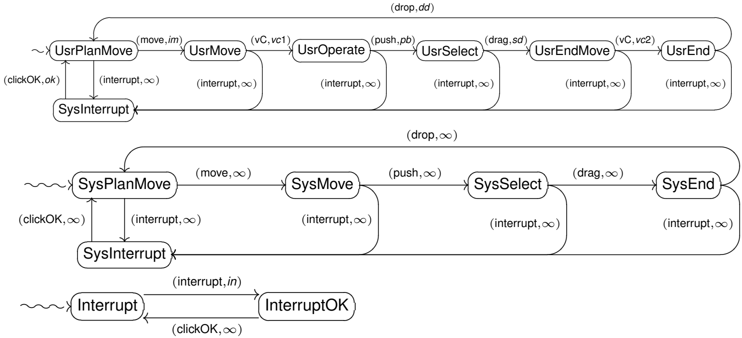
Drag'n'drop model: System



Drag'n'drop model: System



Drag’n’drop model: Interrupt and composition



$$(\text{UsrPlanMove} \bowtie_{\{\text{move,push,drag,drop,interrupt,clickOK}\}} (\text{SysPlanMove} \bowtie_{\{\text{interrupt,clickOK}\}} \text{Interrupt}))$$

Parameter Values for Drag 'n Drop

Rate: 1000 [ms] / mean time [ms]

im = 1000/910;	time of planning (240 ms) plus ballistic (670 ms) movement
vc1 = 1000/290;	time of approach + adjust movement
vc2 = 1000/290;	as above (1000/120 for procedural case)
in ;	interrupt time variable
pb = 1000/120;	time of completion of movement finishing with a push button
sd = 1000/680;	time planning (0) and ballistic (680 ms)
dd = 1000/120;	time to release button and check item dropped (120 ms)
ok = 1000/1300;	time needed to handle pop-up interruption (1300 ms)

Each cognitive transformation in an ICS configuration takes ca. 40 ms

Translation into PRISM: Interrupt model

```
rate in ;
rate ok = 1000/1300;

const Interrupt = 0;
const InterruptOK = 1;

module Interrupt

      Interrupt_STATE : [0..1] init Interrupt;

      [interrupt]
         (Interrupt_STATE = Interrupt)
         -> in :
         (Interrupt_STATE' = InterruptOK);
      [clickOK]
         (Interrupt_STATE = InterruptOK)
         -> ok :
         (Interrupt_STATE' = Interrupt);

endmodule
```

PRISM specification rewards

Reward structures

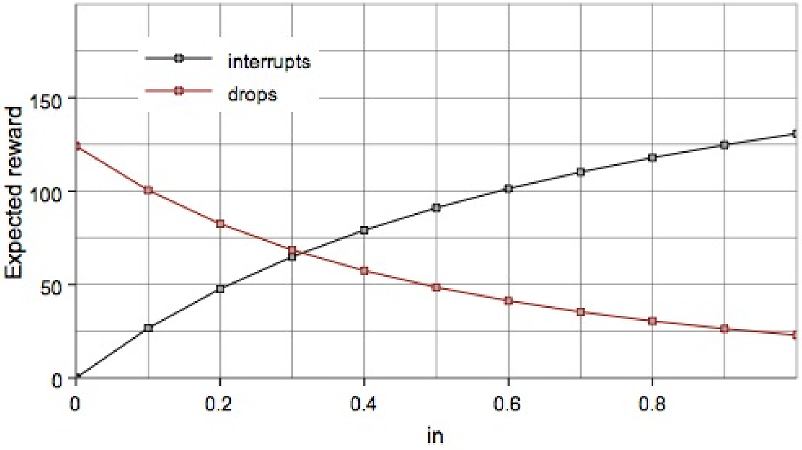
```
rewards "drops"
      [drop] true : 1;
endrewards

rewards "moves"
      [move] true : 1;
endrewards

rewards "interrupts"
      [interrupt] true : 1;
endrewards
```

Reward measure

$$R\{\text{'drops'}\}=?[C<300]$$

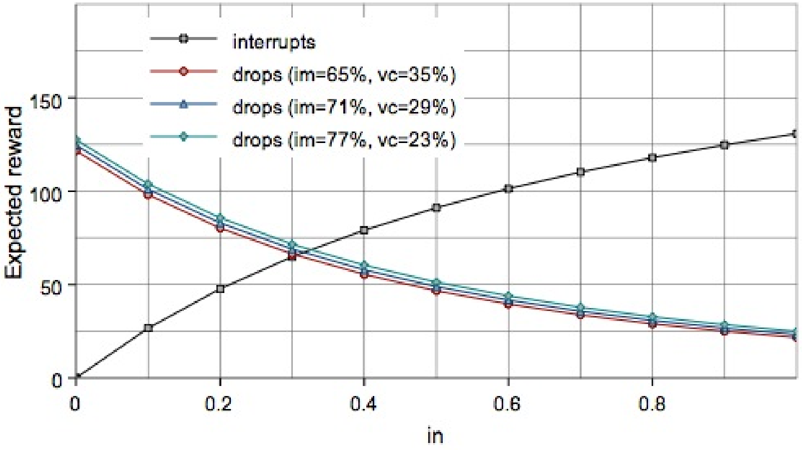
Cumulative number of drop-actions over 300 seconds

Resilience of D'n'D to interruptions



Reward measure: R{'drops'}=?[C<300]
Cumulative number of drop-actions over 300 seconds for D'n'D

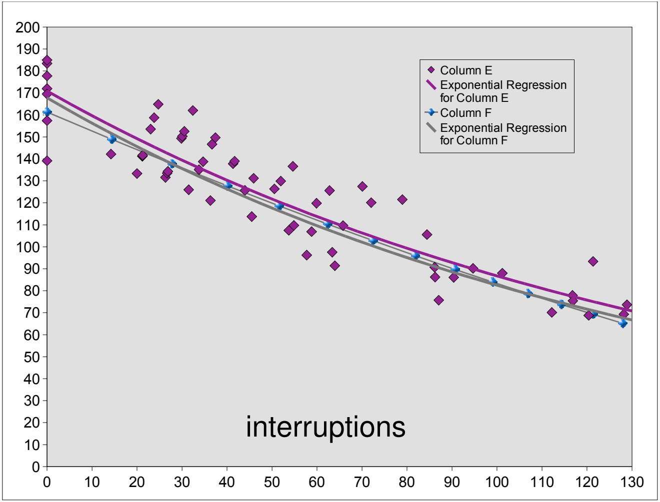
Sensitivity to rate parameters: D’n’D



Cumulative number of drop-actions over 300 seconds

12% variation in movement phases leads to 4% variation in nr. of drops

drops



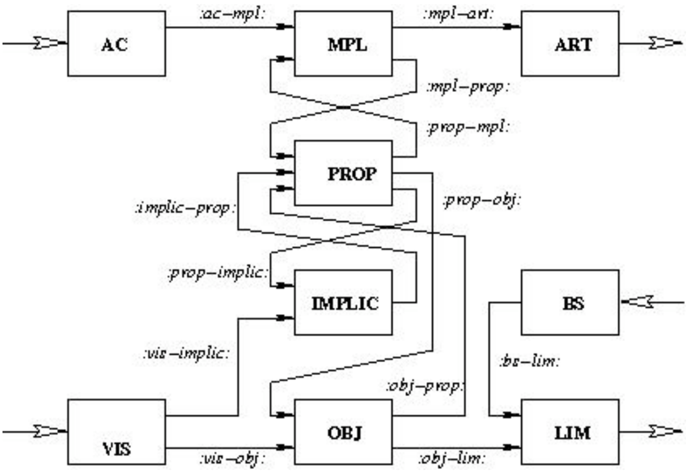
Column E: empirical nr. of drops: 6 subj., 5 min., at nr. interr.

Column F: nr. of drops predicted by model with related par. values

(Cognitive) Activities involved in Speak 'n Drop

- Visual observation of desktop
- Selection of icon to remove (3-4 phases)
- Meanwhile: pronounce 'delete' command

ICS configuration Speak and Drop



ICS and parallel activities

Speaking and pointing are partially in conflict:

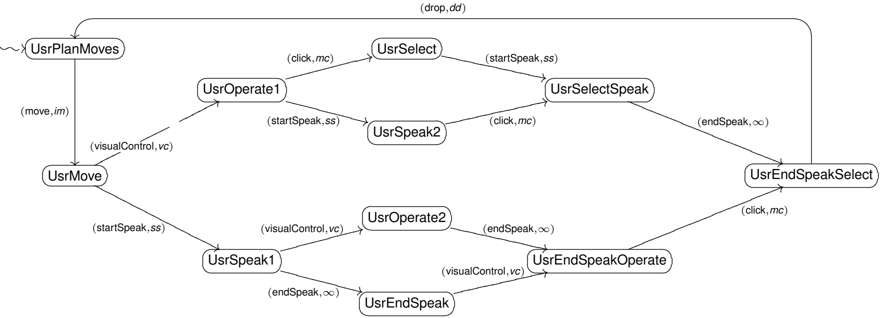
- Visual control requires focus of attention
- Starting to speak requires focus of attention

therefore:

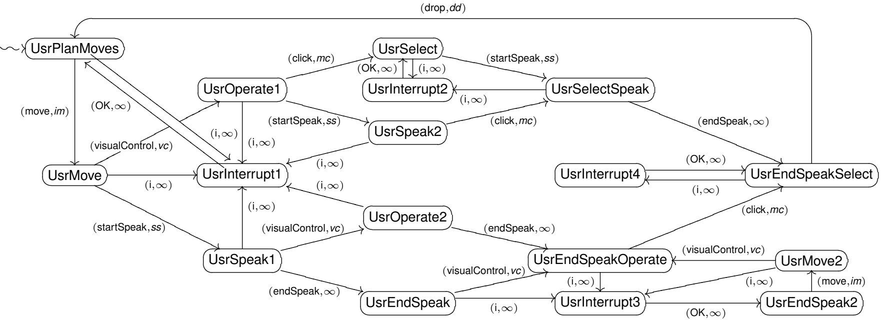
- Start speech after icon is selected XOR
- Start speech during proceduralised ballistic phase

Timing: each cognitive transformation in a configuration takes about 40 ms. So, time to put a configuration into place depends on number of transformations needed. This will be taken care of in the 'planning' phase.

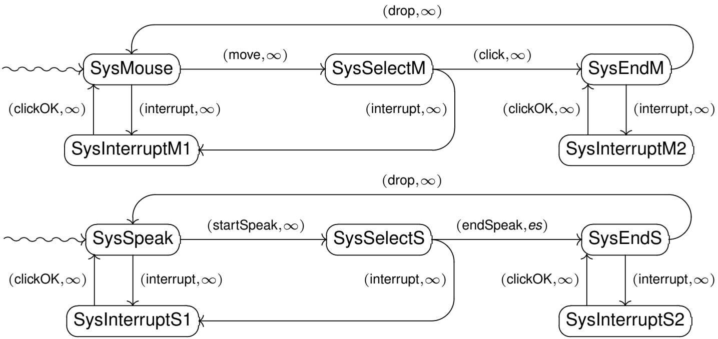
Speak 'n drop model: User



Speak'n'drop model: User



S'n'D model: System & Interruptions



$$((\mathsf{UsrPlanMoves} \bowtie_{\{move,startSpeak,click,endSpeak,drop,interrupt,clickOK\}} (\mathsf{SysMouse} \bowtie_{\{drop,interrupt,clickOK\}} \mathsf{SysSpeak})) \bowtie_{\{interrupt,clickOK\}} \mathsf{Interrupt})$$

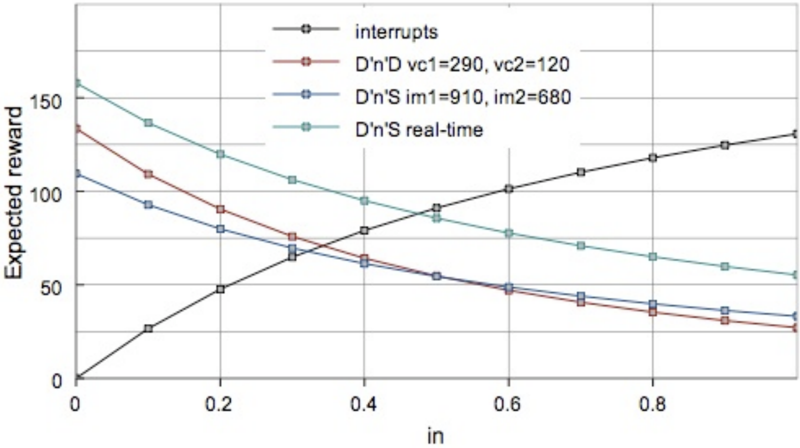
Parameter Values for Speak 'n Drop

Rate: 1000 [ms] / mean time [ms]

S'n'D:

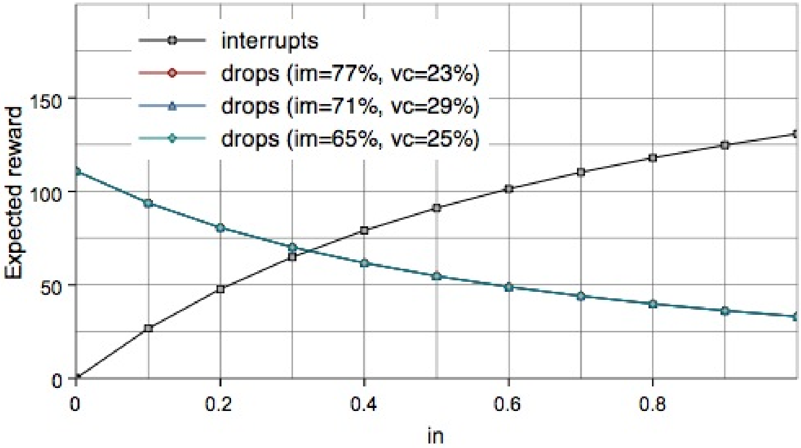
im = 1000/910;	time of planning initial movement plus ballistic movement
vc = 1000/290;	time of visual control
in ;	interrupt time variable
mc = 1000/80;	time of completion of movement finishing with a mouse click
ss = 1000/630;	time for user to start speaking and completing the utterance
es = 1000/1000;	time for user to end speaking (plus recognition and feedback)
dd = 1000/120;	time to check item is removed
ok = 1000/1300;	time to handle pop-up interruption

Resilience of D’n’D vs. S’n’D



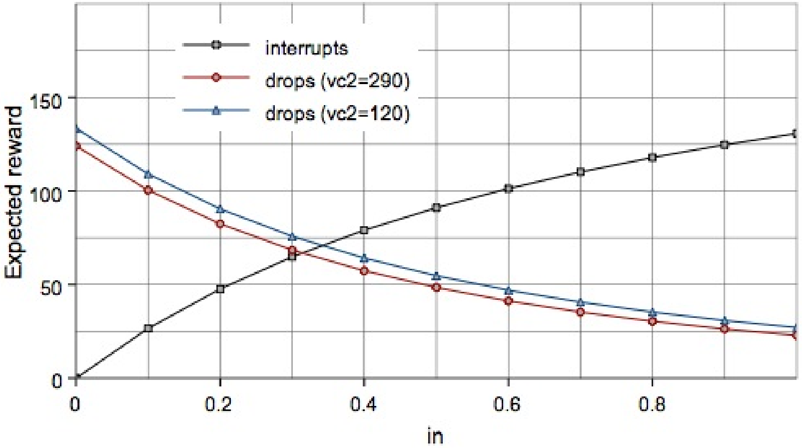
Reward measure: R{’drops’}=?[C<300]
Cumulative number of drop-actions over 300 seconds

Sensitivity to rate parameters: S'n'D



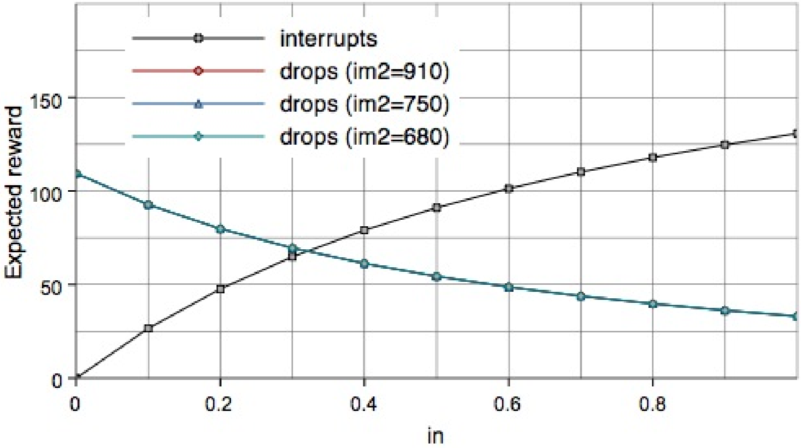
Cumulative number of drop-actions over 300 seconds
12% variation in movement phases leads to 1% variation in nr. of drops

Learning Effect: D’n’D



Cumulative number of drop-actions over 300 seconds
Effect of learning in visual control phase in D’n’D

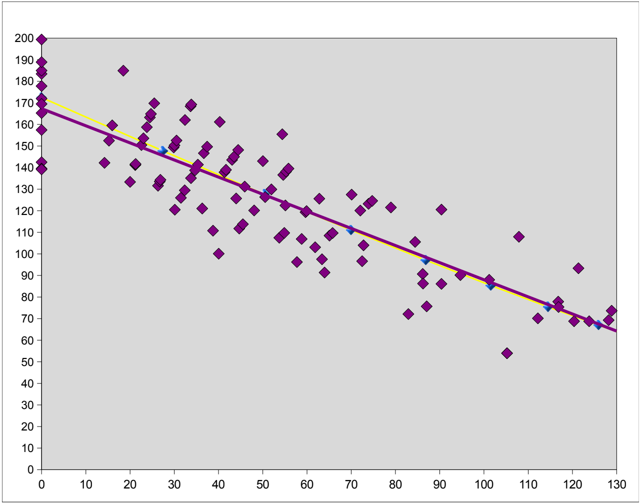
Learning Effect: S'n'D



Cumulative number of drop-actions over 300 seconds
Effect of learning in ballistic phase in S'n'D

Validation: S'n'D (preliminary)

drops



interruptions

Summary and outlook

Stochastic Models for Resilience to Interruptions:

- Encouraging results obtained for D’nD and S’nD
- Validation of results by means of empirical data (ongoing)
- Inclusion of error behaviour and mode confusion
- Study further interaction techniques
- Bridge between ICO/Petri-Nets and stochastic reward model-checking

Literature

ICS:

- P.J. Barnard and J. May, Cognitive modelling for user requirements. In *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services* (P.F. Byerley, P.J. Barnard and J. May, Eds.), Elsevier, 1993, 101–145.
- P.J. Barnard, J. May, D.J. Duke and D.A. Duce, Systems, interactions and macrotheory. *ACM Transactions on Human-Computer Interaction* 7(2), 2000, 222–262.
- D.J. Duke, P.J. Barnard, D.A. Duce and J. May, Syndetic modelling. *International Journal of Human Computer Interaction* 13(4), 1998, 337–393.

Fitts' Law and Human factors:

- G.P. Faconti and M. Massink, Analysis of pointing tasks on a white board. In *Revised Papers of DSV-IS'06*, LNCS 4323, Springer, 2007, 185–198.
- P.M. Fitts, The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology* 47(6), 1954, 381–391.

Literature (cont'd)

Stochastic Model Checking:

- J. Hillston, *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- M. Kwiatkowska, G. Norman and D. Parker. PRISM: Probabilistic Model Checking for Performance and Reliability Analysis. ACM SIGMETRICS Performance Evaluation Review, 36(4), pages 40-45. March 2009. URL `http://www.prismmodelchecker.org/`.

Interruptions:

- D.C. McFarlane, Coordinating the interruption of people in human-computer interaction. In *Proceedings INTERACT'99*, IOS Press, 1999, 295–303.
- J.G. Trafton and C.A. Monk, Task Interruptions. *Reviews of Human Factors and Ergonomics* 3, 2007, 111–126.
- Beek ter M., Faconti G., Massink M., Palanque P., Winckler M. Resilience of Interaction Techniques to Interruptions. In: *Proceedings INTERACT 2009 - Human-Computer Interaction.* 12th IFIP TC 13 International Conference (Uppsala, Sweden, 24-28 August 2009). Proceedings, vol. I pp. 494 - 509. (Lecture Notes in Computer Science, vol. 5726). Springer Verlag, 2009.