University of Glasgow | School of Computing Science

Probabilistic Formal Analysis of App Usage to Inform Redesign

Oana Andrei
with Muffy Calder, Matthew Chalmers, Alistair Morrison, Mattias Rost

School of Computer Science Seminar, University of St Andrews
9 February 2016

Installing and using an app

Installing and using an app

Installing and using an app

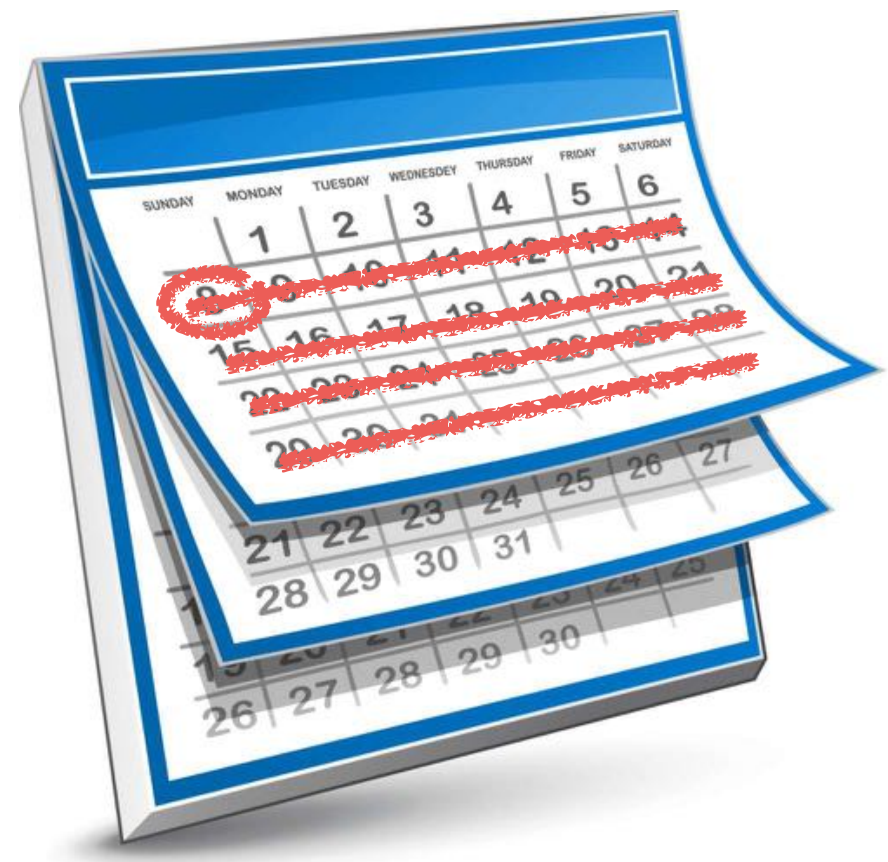
Installing and using an app

Download on the App Store

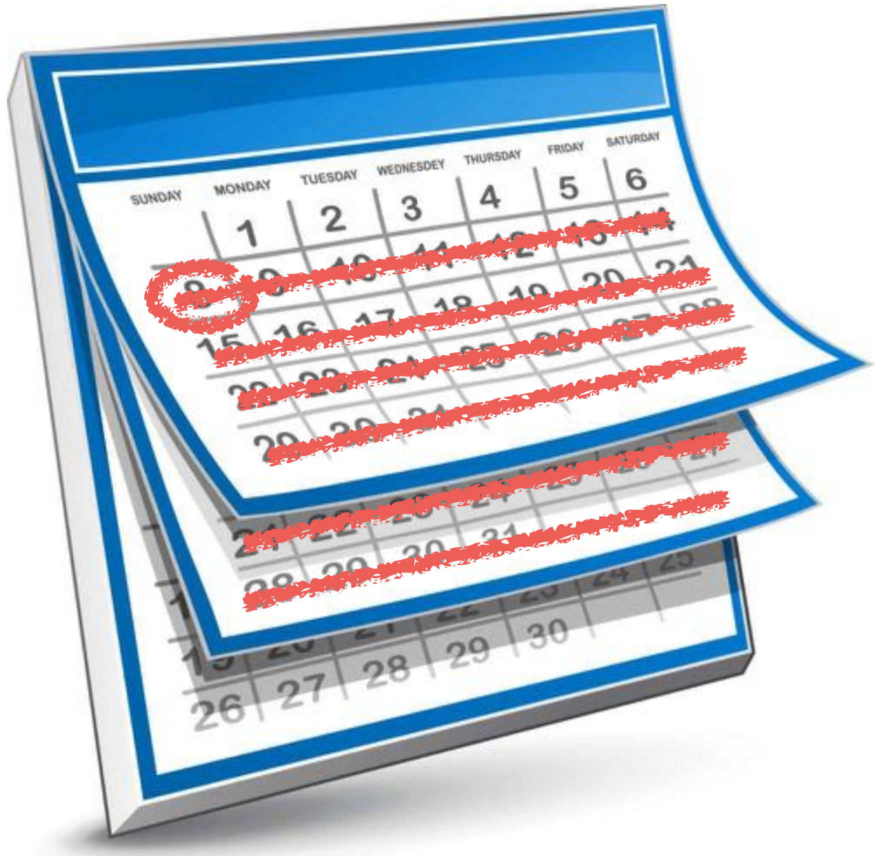
ANDROID APP ON Google play

Installing and using an app

Installing and using an app



Installing and using an app

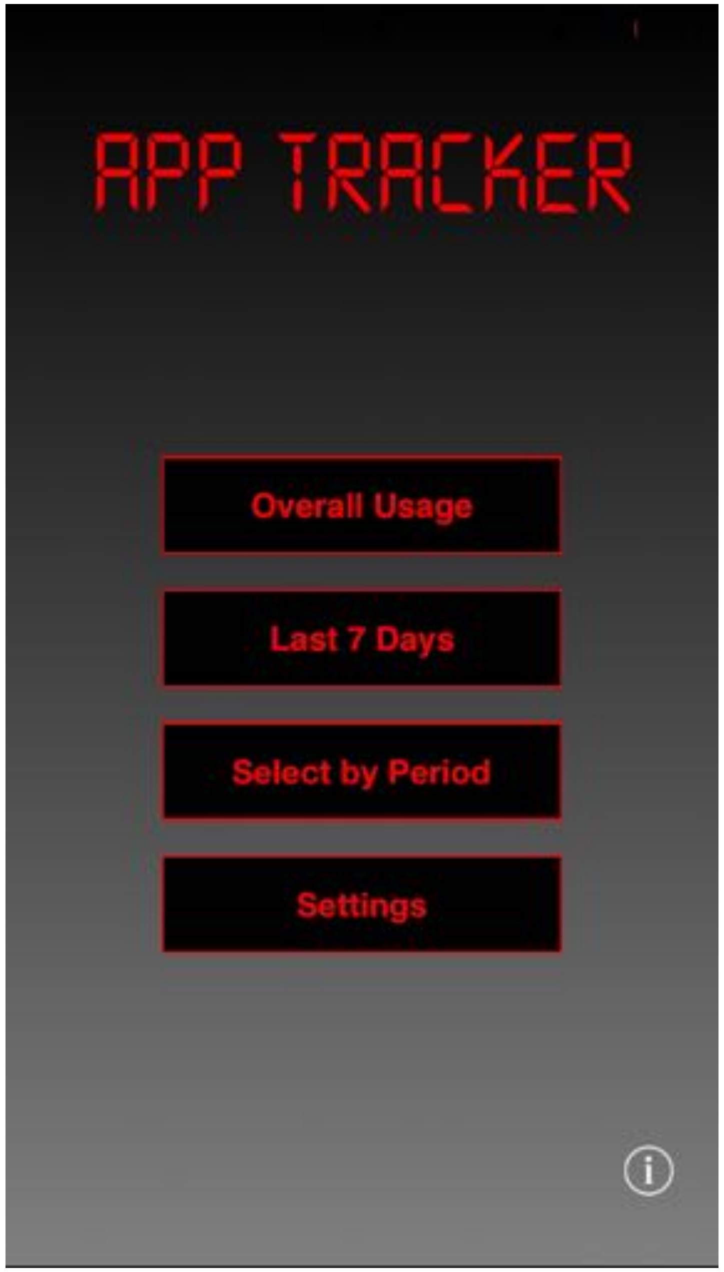


Our motivation

- Users engage with an app in different ways — understand them and use them to inform the app redesign.
- How should we identify and characterise the different styles of use within a population of users?
- How does such characterisation evolve:
    - over an individual user trace?
    - over a number of sessions?
    - over days and months?

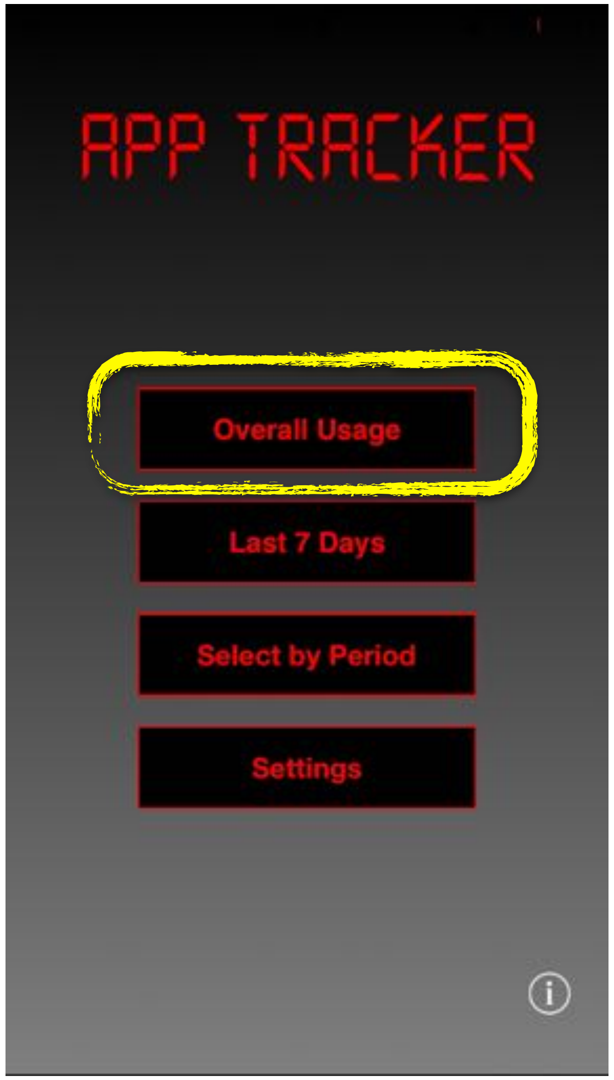
Case study: the **AppTracker** app

- Runs in the background
- Records opening and closing of apps, locking and unlocking the device
- Provides charts and statistics about the device usage
- Over 35K downloads



AppTracker main menu

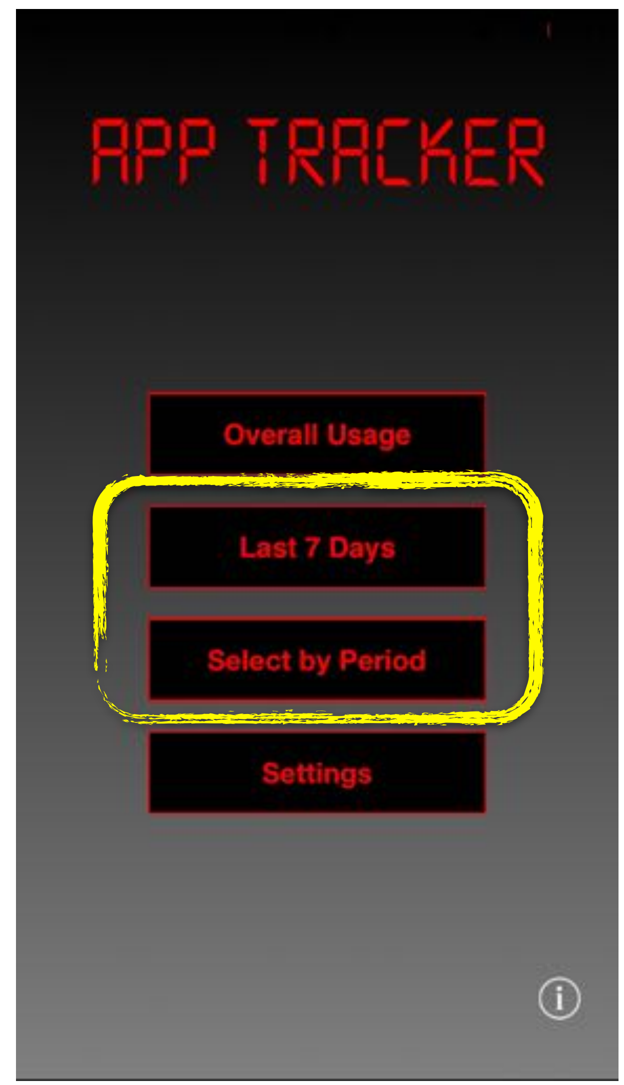
- ***Overall Usage*** provides a summary of all the data recorded since AppTracker was installed:
    - ✦ ***Most Used Apps (Top Apps)***
    - ✦ ***Stats***

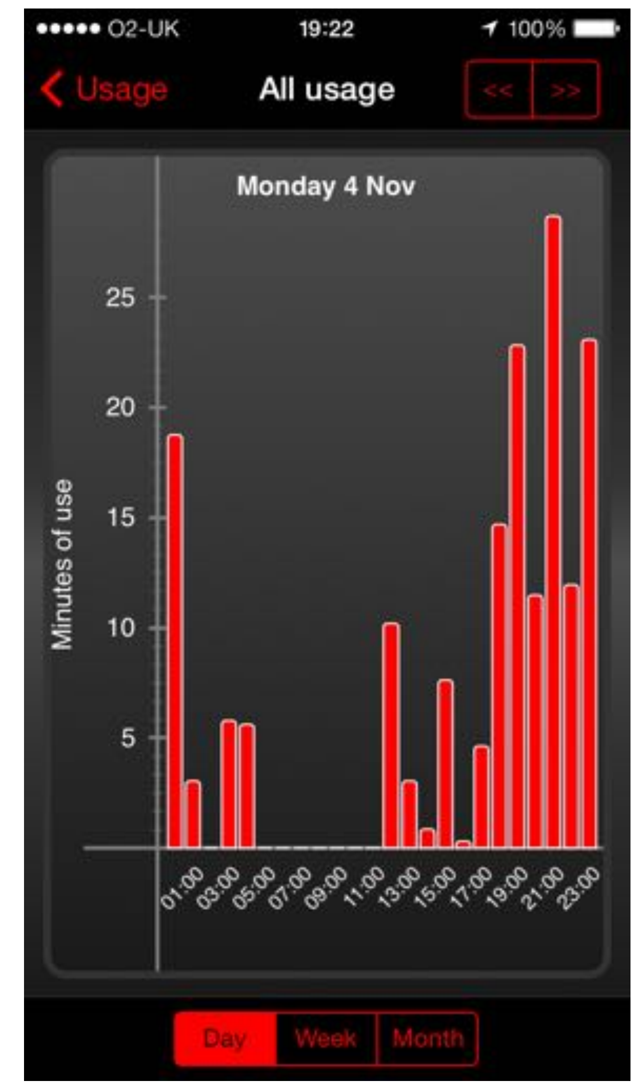


AppTracker main menu

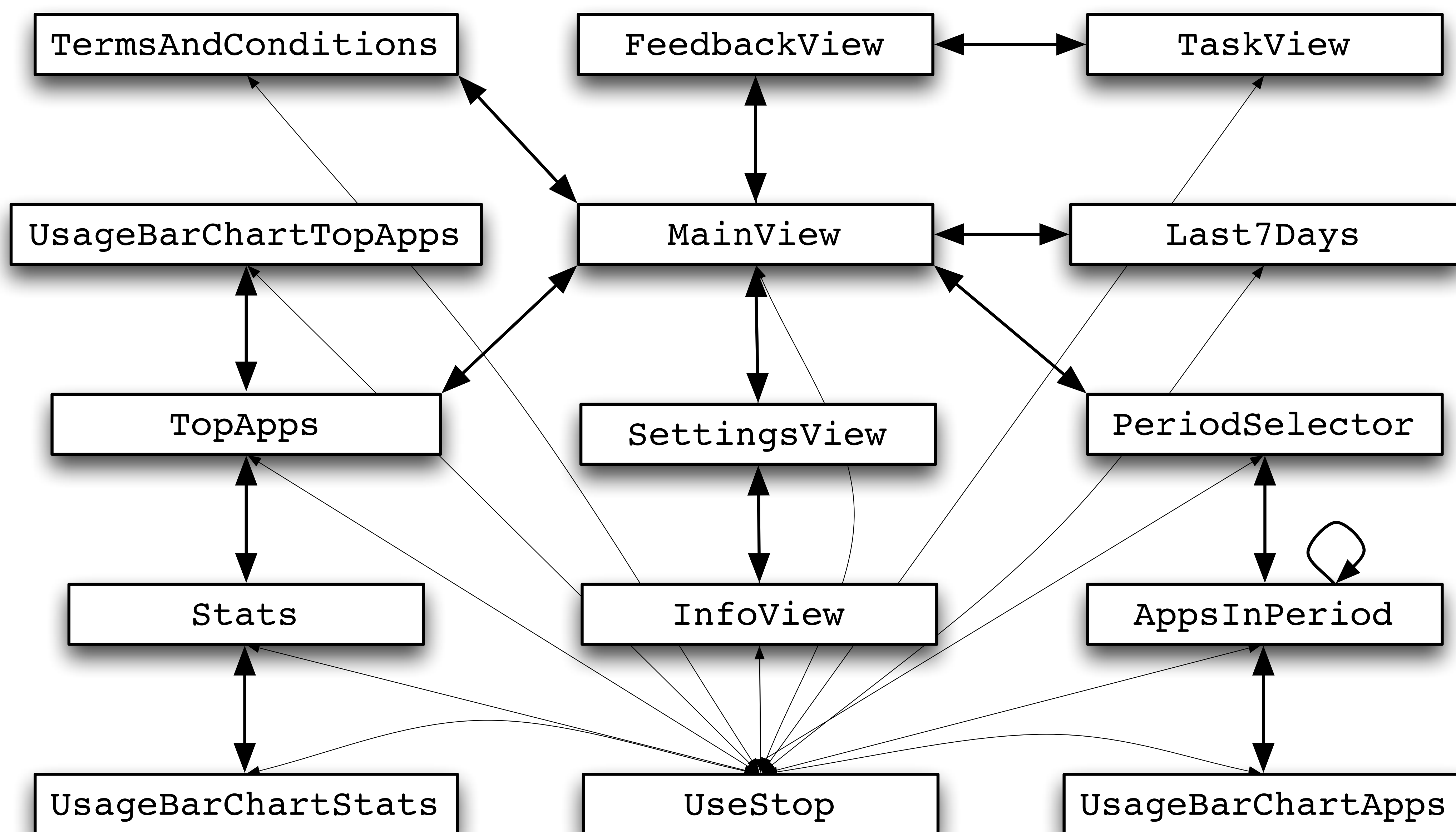
- ***Last 7 Days*** shows a stacked bar graphs of usage of the top 5 apps during the last 7 days of usage.

- ***Select by Period*** shows statistics by any period of interest, e.g.
  - ✦ most used app last Monday
  - ✦ time spent on Facebook last week
  - ✦ device usage over a day

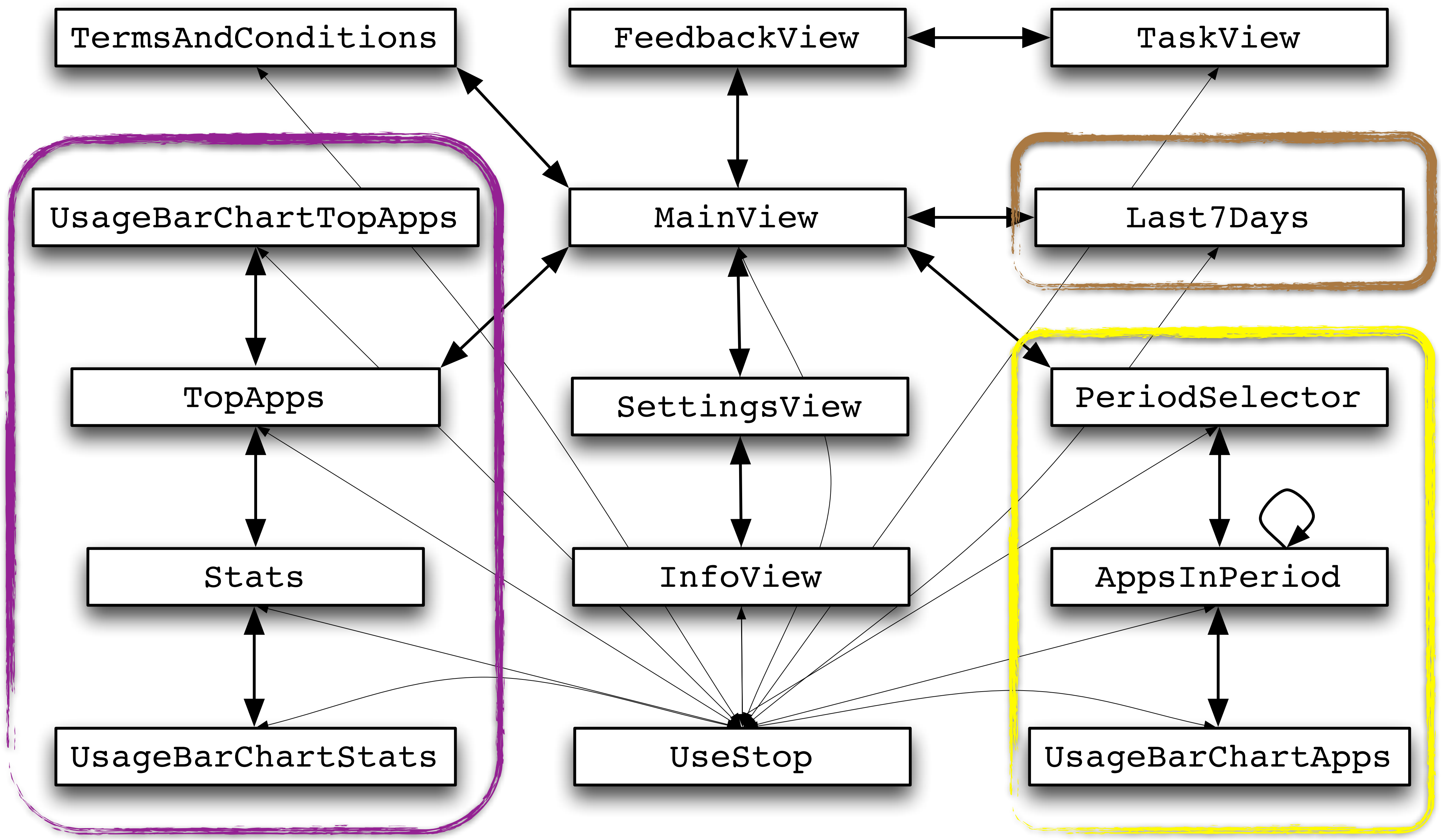




AppTracker state diagram



AppTracker hypothesised behaviour (frequent)



Overview of the analysis

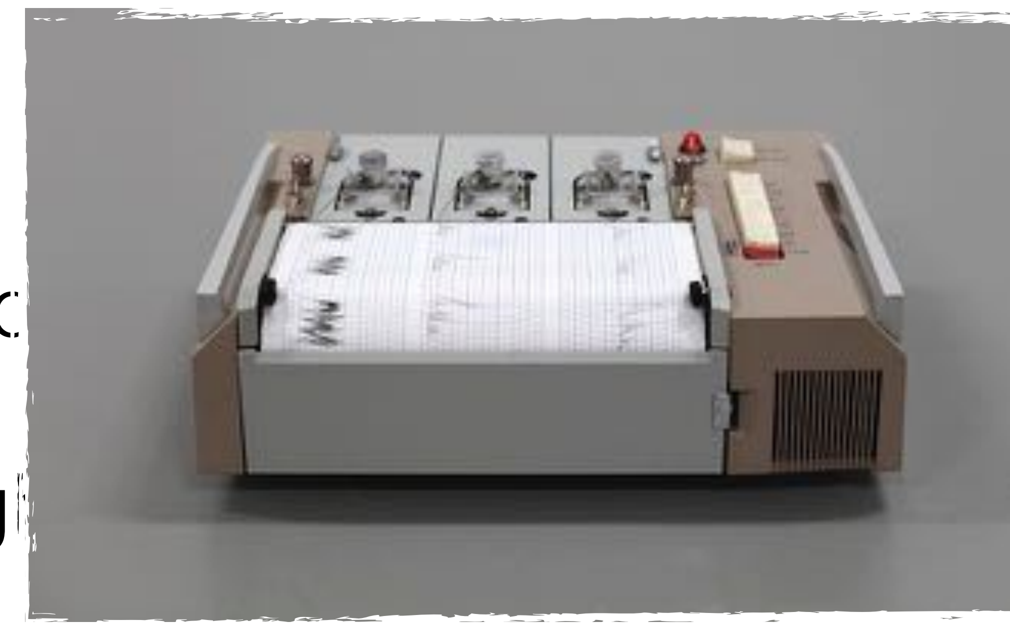
- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings for given parameters and data sets
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- Compare user weightings / patterns distributions
- Discuss with developers to inform redesign

Overview of the analysis

- **Instrument the app**

- Clean and prepare the raw logged d

- Infer activity patterns and user weig
  parameters and data sets



- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM

- Compare user weightings / patterns distributions

- Discuss with developers to inform redesign

Overview of the analysis

- Instrument the app
- **Clean and prepare the raw logged data**
- Infer activity patterns and user weightings for given parameters and data sets
- Ask questions about the pattern
temporal properties and model
- Compare user weightings / patt
- Discuss with developers to info

Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- **Infer activity patterns and user weightings for given parameters and data sets**
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- Compare user weightings / patterns distributions
- Discuss with developers to inform redesign

Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings for given parameters and data sets
- **Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM**
- Compare user weightings / patterns distributions
- Discuss with developers to inform redesign

Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings parameters and data sets
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- **Compare user weightings / patterns distributions**
- Discuss with developers to inform redesign

Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings for given parameters and data sets
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- Compare user weightings / patterns distributions
- **Discuss with developers to inform redesign**

Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings for given parameters and data sets
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- Compare user weightings / patterns distributions
- Discuss with developers to inform redesign

Not sequential!

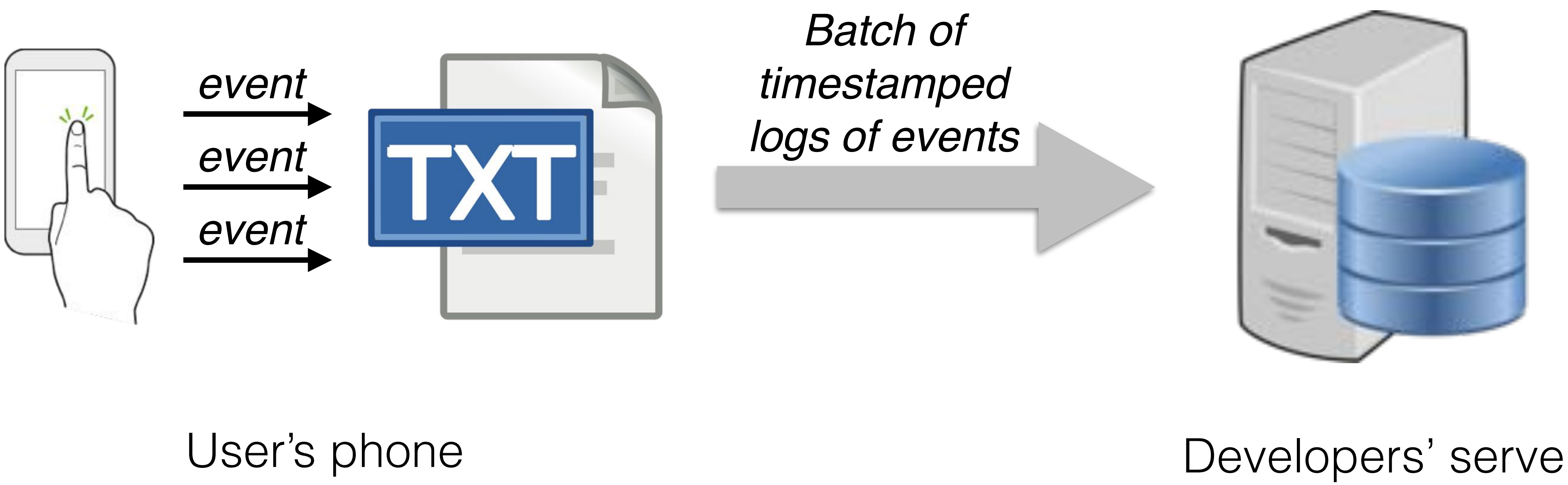
Overview of the analysis

- Instrument the app
- Clean and prepare the raw logged data
- Infer activity patterns and user weightings for given parameters and data sets
- Ask questions about the patterns using probabilistic temporal properties and model checking in PRISM
- Compare user weightings / patterns distributions
- Discuss with developers to inform redesign

Complementary to other data analytics methods

(e.g., SQL queries, visualisations)

Instrument the app



SGLog data logging infrastructure (SUM group@GU)

Prepare the raw logged data

- User traces based on 15 selected state abstractions:


```
[{"deviceid":"xx:xx:xx:xx:xx:xx","totalevents":230,"firstSeen":"2013-08-20
09:10:59","lastSeen":"2014-03-24 09:57:32","sessions":[[{"timestamp":"2013-08-20
09:11:02","data":"TermsAndConditions"},{"timestamp":"2013-08-20 09:11:23",
"data":"Main"},{"timestamp":"2013-08-2009:11:46","data":"TopApps"},
{"timestamp":"2013-08-20 09:11:50”,"data":"Main"},
{"timestamp":"2013-08-2009:11:52","data":"Last7Days"},{"timestamp":"2013-08-20
09:11:56", "data":"Main"},{"timestamp":"2013-08-20:11:59", "data":"PeriodSelector"},
{"timestamp":"2013-08-20 09:12:04","data":"Main"},{"timestamp":"2013-08-20
09:12:06","data":"UseStop"}],...
```


- Clean up the data: 489 user traces between Aug. 2013 - May 2014
- Segment the session data: intervals of days of usage [0,1), [1,7), [7,30), [0,30), [30,60), [60,90)
- Compute the 15x15 transition-occurrence matrix for each trace in a given data set

Infer activity patterns

- Look for $\mathcal{K}$ distinct behaviours
  - Run a non-linear optimisation algorithm for parameter estimation to learn $\mathcal{K}$ admixture bigram models from transition-occurrences matrices
  - $\mathcal{K}$ discrete-time Markov chains Φ_k — **activity patterns**: $\Phi_k[i,j]$ is the probability of moving from state i to state j while in Φ_k
  - for each user trace, a weight vector $(\Theta_1, \ldots, \Theta_{\mathcal{K}})$ with Θ_k the probability of using the k^{th} activity pattern

About PRISM

Probabilistic model checker (Birmingham & Oxford)

- probabilistic models expressed in a high-level state-based language (DTMC, MDP, CTMC, etc.)
- model checking quantitative properties expressed as temporal logic formulae (PCTL, CSL, PCTL*, etc.), extensions with costs/rewards
- exhaustive analysis of all possible executions of the model

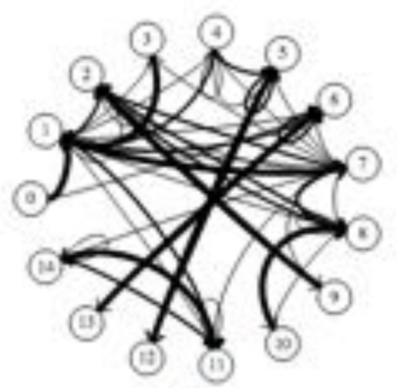
Activity patterns in PRISM

- Generate a PRISM model for each activity pattern
  - 15 states — one for each view, including `UseStop`
  - reward structures for:
    - visiting a specific screen view (state) — reward value 1
    - counting button taps (steps/transitions) — reward value 1
- What can we say about each activity pattern?

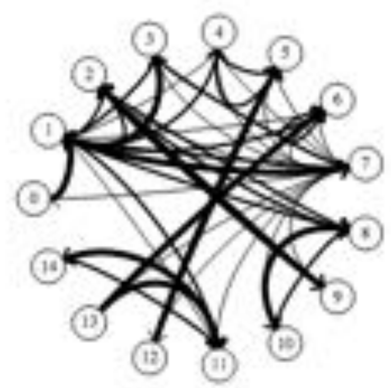
Activity patterns for the first 30 days

Activity patterns for the first 30 days

2 patterns



AP1



AP2

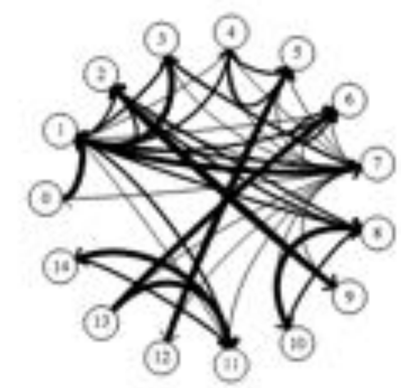
Discrete-time Markov chains: the arrows are probabilistic transitions.

Activity patterns for the first 30 days

Discrete-time Markov chains: the arrows are probabilistic transitions.

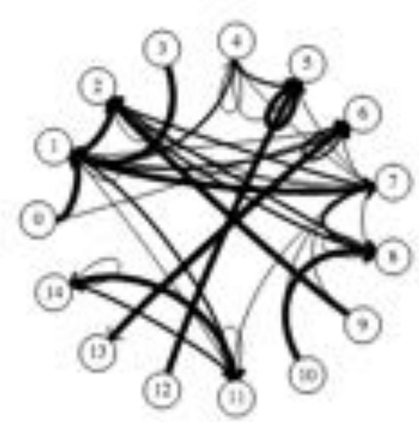
2 patterns

AP1

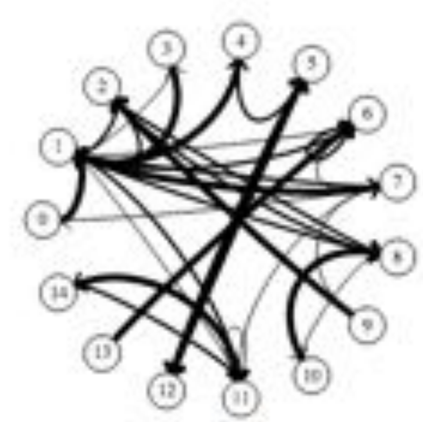


AP2

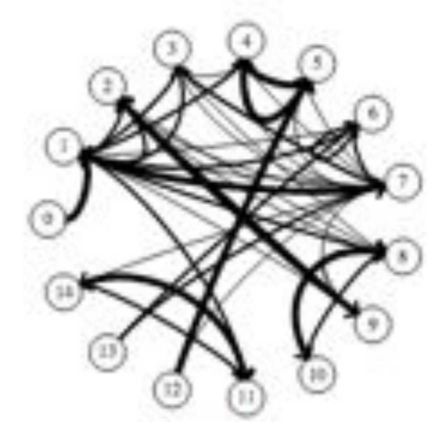
3 patterns



AP1



AP2

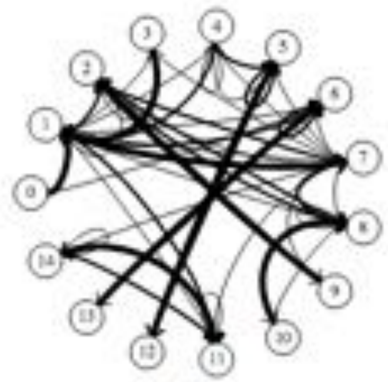


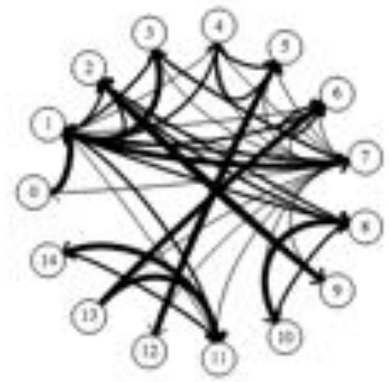
AP3

Activity patterns for the first 30 days

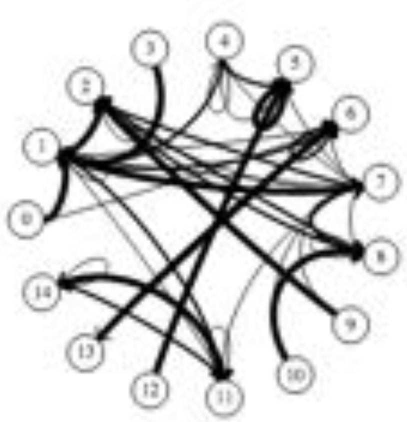
Discrete-time Markov chains: the arrows are probabilistic transitions.

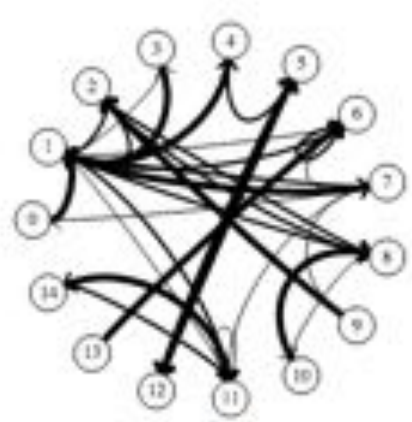
2 patterns

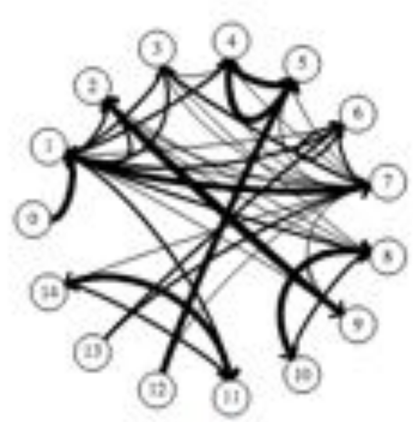

AP1


AP2

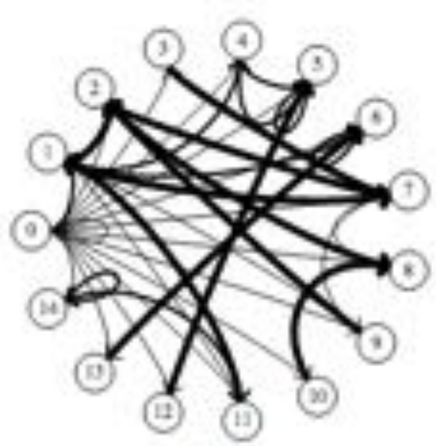
3 patterns

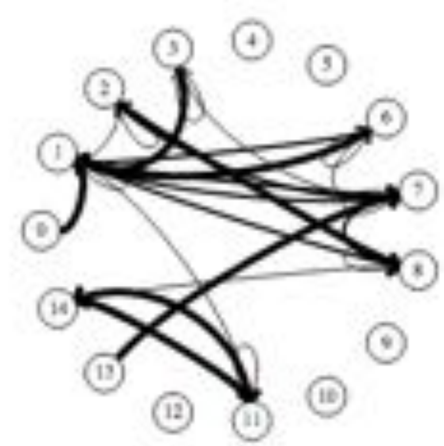

AP1

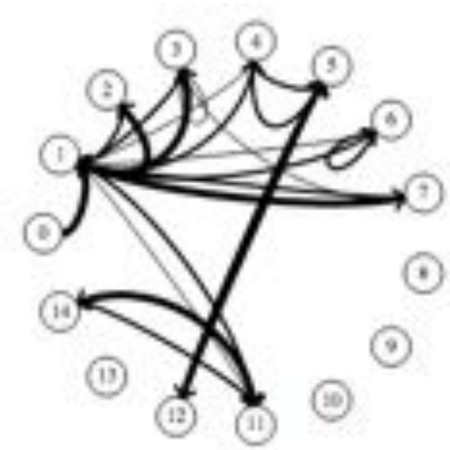

AP2

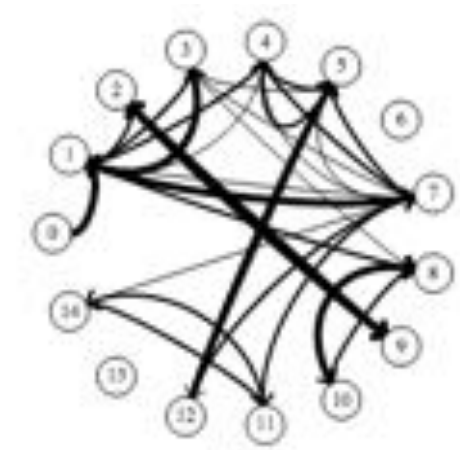

AP3

4 patterns


AP1


AP2


AP3


AP4

What questions can we ask?

- An exploratory process of identifying the “good” questions:
  - ✦ any type of app
  - ✦ a particular type of app (e.g. goal-oriented apps)
  - ✦ a particular app
- Find experiential questions in order to identify:
  - ✦ more relevant questions to ask and
  - ✦ most relevant states to query

Formulate temporal properties

(PCTL with rewards)

- General questions:
  - Likelihood of viewing a particular screen for the first time within 100 taps.
  - Average number of views of a particular screen within 20 button taps.
  - Average number of button taps to reach a particular screen view, etc.
- More app-specific questions:
  - Probability to perform an event if always reading ***InfoView*** within 25 steps.
  - Average number of button taps to go to screen view s_2 from s_1.
  - Probability of repeating a specific event *50* times, etc.

Formulate temporal properties

(PCTL with rewards)

- General questions:
    - Likelihood of viewing a particular screen for the first time within 100 taps.
    - Average number of views of a particular screen within 20 button taps.
    - Average number of button taps to reach a particular screen view, etc.
- More app-specific questions:
    - Probability to perform an event if always reading ***InfoView*** within 25 steps.
    - Average number of button taps to go to screen view s_2 from s_1.
    - Probability of repeating a specific event *50* times, etc.

Compare the results across all patterns!

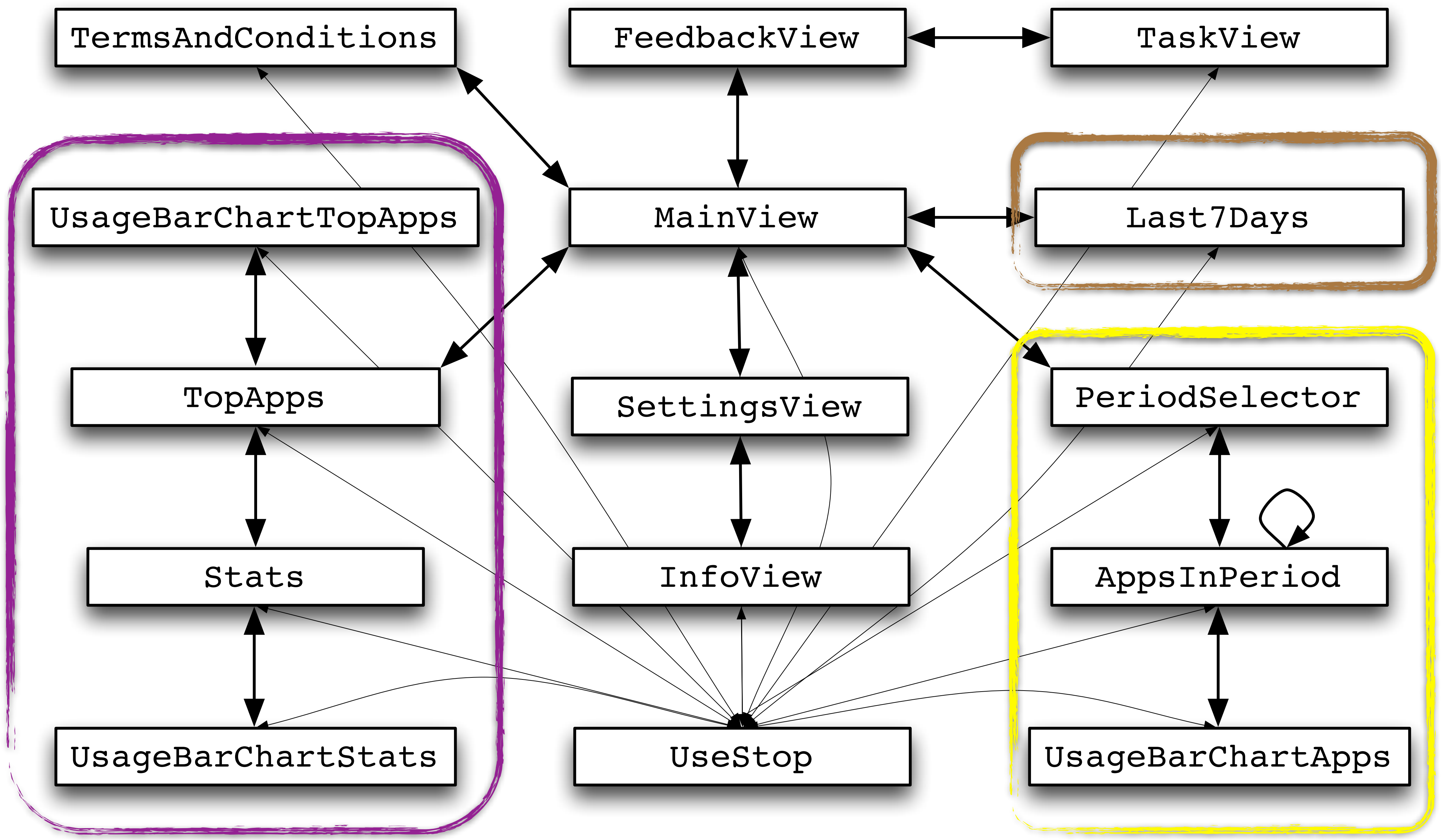
Formulate temporal properties

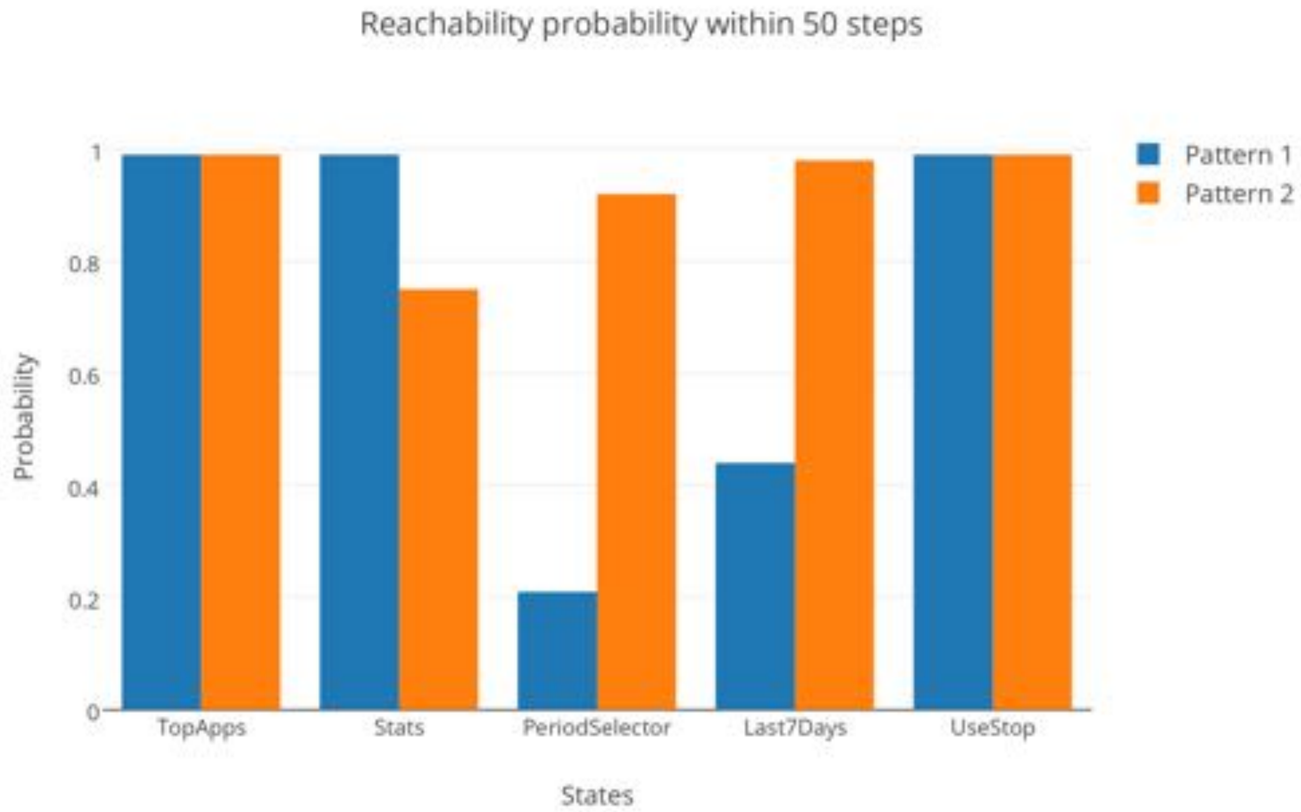
- Probability to reach the state **`s`** for the first time within **`N`** steps:
  - `P=? [!s U<=N s]`
- Expected number of visits to the state **`s`** within **`N`** steps:
  - `R{“r_s”}=? [C<=N]`
- Expected number of steps to reach the state **`s`**:
  - `R{“r_Steps”}=? [F s]`

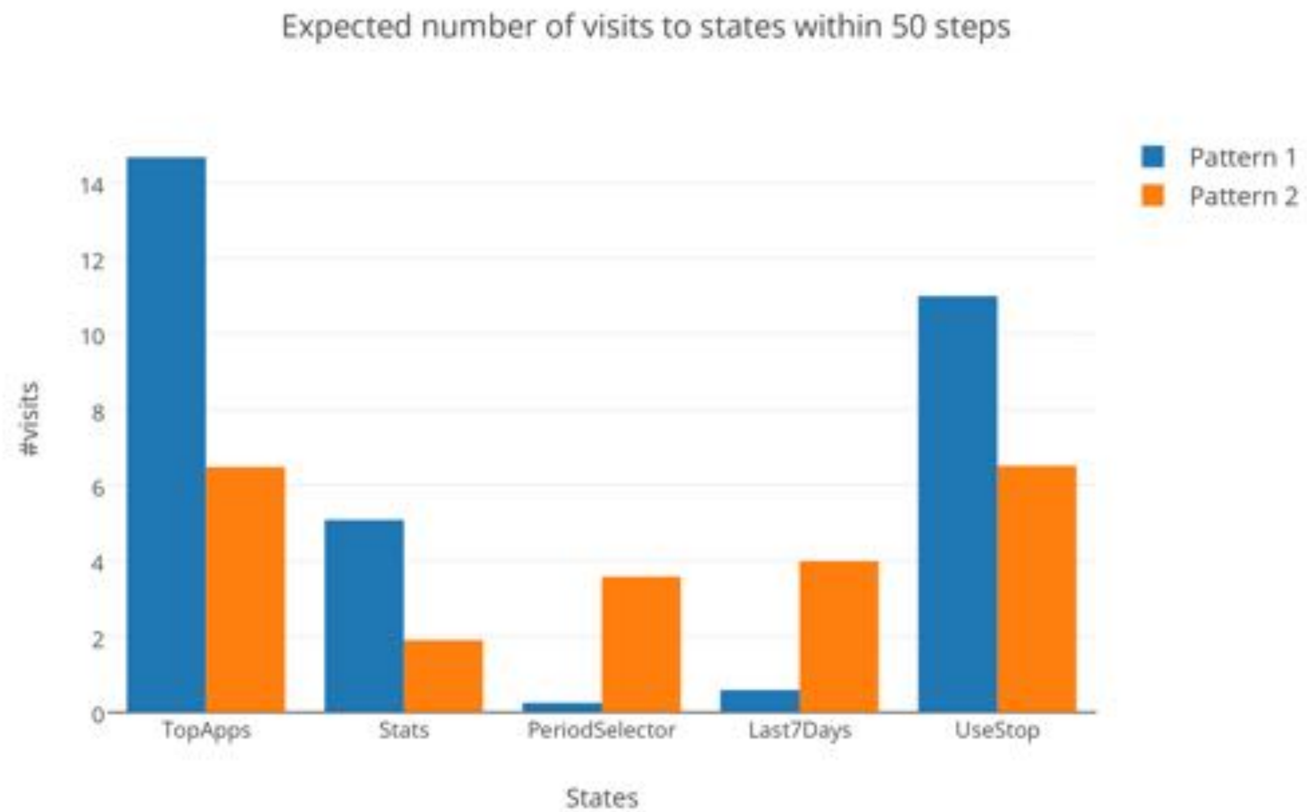
Probabilistic model checking

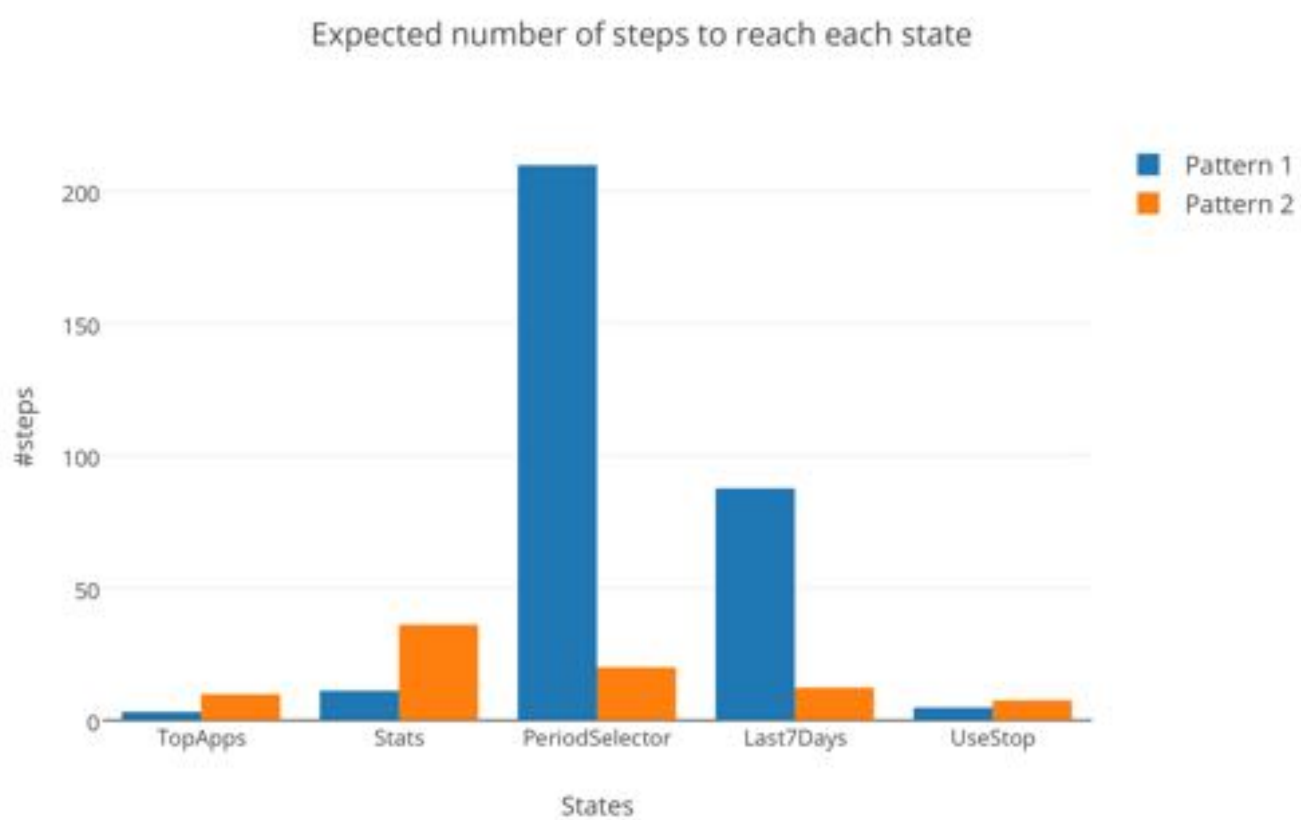
- Probabilistic Computation Tree Logic (PCTL) with rewards in PRISM
- Compare the results across:
    - all activity patterns,
    - states: `TopApps, Stats, PeriodSelector, Last7Days, UseStop,`
    - intervals of days of usage [0,1), [1,7), [7,30), [0,30), [30,60), [60,90)

AppTracker hypothesised behaviour









First month analysis:

- two activity patterns ($\mathcal{K}$=2)
- Pattern 1
  - Higher `TopApps` and `Stats`
  - Shorter and more frequent sessions
- Pattern 2
  - Significant results for `Last7Days` and `PeriodSelector`
  - Longer and less frequent sessions

Table 2: **Property 1** (the probability of reaching a given state for the first time within N steps), **Property 2** (the expected number of visits to a given state within N steps), and **Property 3** (the expected number of time steps to reach a given state) checked for different states and time cuts, and for $N = 50$ steps

Prop.	Time cut	TopApps		Stats		PeriodSelector		Last7Days		UseStop	
		AP1	AP2	AP1	AP2	AP1	AP2	AP1	AP2	AP1	AP2
Property 1	$[0, 1)$	0.99	0.99	0.99	0.83	0.47	0.79	0.49	0.96	0.99	0.99
	$[1, 7)$	0.99	0.99	0.98	0.80	0	0.93	0	0.98	0.99	0.99
	$[7, 30)$	0.99	0.99	0.99	0.64	0.01	0.94	0.84	0.96	0.99	0.99
	$[0, 30)$	0.99	0.99	0.99	0.75	0.21	0.92	0.44	0.98	0.99	0.99
	$[30, 60)$	0.99	0.99	0	0.90	0.73	0.83	0.56	0.98	1	0.99
	$[60, 90)$	1	0.95	0.96	0.72	0	0.94	0	0.97	1	0.99
Property 2	$[0, 1)$	13.94	7.44	7.63	2.15	0.79	1.82	0.70	3.13	11.41	6.17
	$[1, 7)$	17.22	5.77	4.00	2.31	0	3.97	0	4.03	12.91	6.30
	$[7, 30)$	14.93	7.15	5.43	1.47	0.01	4.61	1.78	3.41	12.86	5.74
	$[0, 30)$	14.67	6.48	5.08	1.90	0.24	3.58	0.58	3.99	11.00	6.51
	$[30, 60)$	13.40	6.83	0	3.76	4.41	2.04	0.85	4.54	12.46	5.61
	$[60, 90)$	17.30	5.83	2.94	2.60	0	3.26	0	4.43	13.96	5.63
Property 3	$[0, 1)$	3.31	8.41	8.18	28.67	79.32	32.46	74.87	15.56	4.86	7.88
	$[1, 7)$	2.05	10.70	12.44	31.90	∞	19.12	∞	12.38	3.85	7.55
	$[7, 30)$	2.52	9.68	9.70	48.61	∞	17.78	26.61	14.58	3.88	8.44
	$[0, 30)$	3.05	9.73	11.01	36.03	209.68	19.94	87.54	12.19	4.67	7.43
	$[30, 60)$	4.04	10.34	∞	22.33	38.21	28.28	61.74	11.08	1	8.82
	$[60, 90)$	2.02	15.28	16.53	39.68	∞	17.41	∞	11.56	3.57	8.90

Table 2: Property 1 (the probability of reaching a given state for the first time within N steps), Property 2 (the expected number of visits to a given state within N steps), and Property 3 (the expected number of time steps to reach a given state) checked for different states and time cuts, and for $N = 50$ steps

Prop.	Time cut	TopApps		Stats		PeriodSelector		Last7Days		UseStop	
		AP1	AP2	AP1	AP2	AP1	AP2	AP1	AP2	AP1	AP2
Property 1	$[0,1)$	0.99	0.99	0.99	0.83	0.47	0.79	0.49	0.96	0.99	0.99
	$[1,7)$	0.99	0.99	0.98	0.80	0	0.93	0	0.98	0.99	0.99
	$[7,30)$	0.99	0.99	0.99	0.64	0.01	0.94	0.84	0.96	0.99	0.99
	$[0,30)$	0.99	0.99	0.99	0.75	0.21	0.92	0.44	0.98	0.99	0.99
	$[30,60)$	0.99	0.99	0	0.90	0.73	0.83	0.56	0.98	1	0.99
	$[60,90)$	1	0.95	0.96	0.72	0	0.94	0	0.97	1	0.99
Property 2	$[0,1)$	13.94	7.44	7.63	2.15	0.79	1.82	0.70	3.13	11.41	6.17
	$[1,7)$	17.22	5.77	4.00	2.31	0	3.97	0	4.03	12.91	6.30
	$[7,30)$	14.93	7.15	5.43	1.47	0.01	4.61	1.78	3.41	12.86	5.74
	$[0,30)$	14.67	6.48	5.08	1.90	0.24	3.58	0.58	3.99	11.00	6.51
	$[30,60)$	13.40	6.83	0	3.76	4.41	2.04	0.85	4.54	12.46	5.61
	$[60,90)$	17.30	5.83	2.94	2.60	0	3.26	0	4.43	13.96	5.63
Property 3	$[0,1)$	3.31	8.41	8.18	28.67	79.32	32.46	74.87	15.56	4.86	7.88
	$[1,7)$	2.05	10.70	12.44	31.90	∞	19.12	∞	12.38	3.85	7.55
	$[7,30)$	2.52	9.68	9.70	48.61	∞	17.78	26.61	14.58	3.88	8.44
	$[0,30)$	3.05	9.73	11.01	36.03	209.68	19.94	87.54	12.19	4.67	7.43
	$[30,60)$	4.04	10.34	∞	22.33	38.21	28.28	61.74	11.08	1	8.82
	$[60,90)$	2.02	15.28	16.53	39.68	∞	17.41	∞	11.56	3.57	8.90

Probabilistic model checking

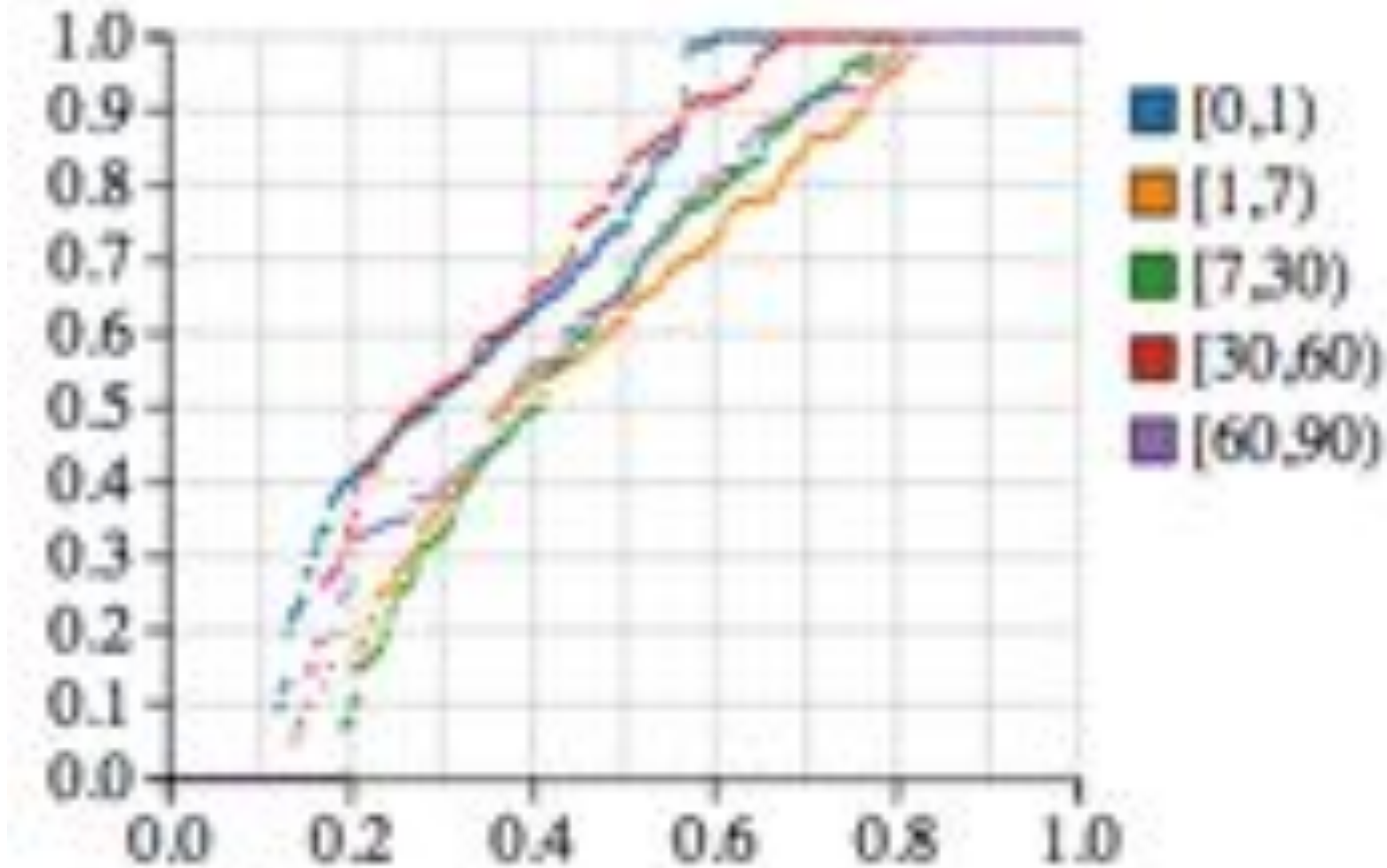
- For the second month data, [30,60), we need more discriminatory properties
- More app-specific questions:
  - Probability to reach `s` from `t` during the same session:
    - `filter(state, P=?[(!s & !"UseStop") U<=N s], t)`
      - for `(s, t)` ∈ $\{$`TopApps, PeriodSelector, Last7Days`$\}^2$
  - Expected number of steps to reach `s` from `t`:
    - `filter(state, R{"r_steps"}=?[F s], t)`
      - for `(s, t)` ∈ $\{$`TopApps, PeriodSelector, Last7Days`$\}^2$
      - for `(s, t)` ∈ `{TopApps, PeriodSelector, Last7Days}` × `Main`
      - for `(s, t)` ∈ `UseStop` × `{TopApps PeriodSelector, Last7Days}`

Probabilistic model checking

- Conclusions for $\mathcal{K}$=2 — two distinct activity patterns labelled:
    - Overall viewing pattern (pattern 1) — higher level stats visualisations
    - Time-partitioned viewing pattern (pattern 2) — in-depth stats visualisations

Comparing pattern distributions $\mathcal{K}$=2

- First day dominated by app exploration
- Interval [30,60) sees a rise in app exploration
- Intervals [1,7), [7,30), [60,90) — more settled usage behaviour



Probability of a user trace to behave according to the Time-partitioned viewing pattern

Analysis for $\mathcal{K}$=3, first month data

1. Overall Viewing pattern:
   - `TopApps` and `Stats` have best results for all three general properties,
   - `PeriodSelector` and `Last7Days` are absent,
   - twice as short and twice more frequent sessions than for pattern #3.
2. 'weaker' Overall Viewing pattern than pattern #1:
   - `TopApps` has poorer results than #1, and better results than `Stats` and `Last7Days` in #2,
   - `PeriodSelector` is absent.
3. Time-partitioned Viewing pattern:
   - `PeriodSelector` has the best results, followed closely by `TopApps` and `Last7Days`.

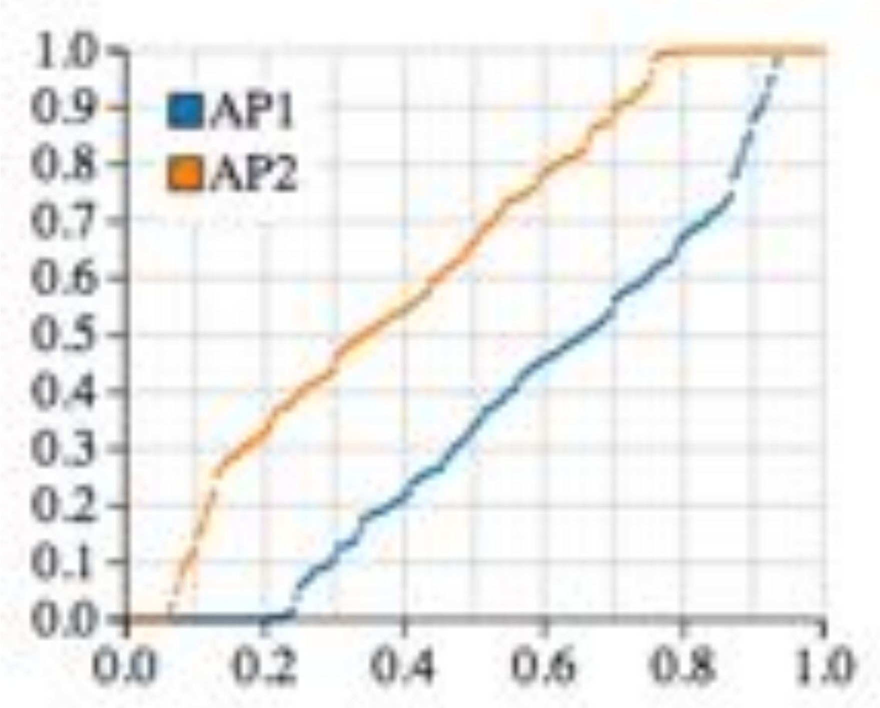
Analysis for $\mathcal{K}$=4, first month data

- Activity patterns:
  1. mainly **`TopApps`** Viewing
  2. mainly **`Stats`** — **`TopApps`** Viewing
  3. Time-Partitioned Viewing
  4. exclusive **`TopApps`** — **`UsageBarChartTopApps`**
- Shorter and more frequent sessions for #1 than for #2 and #3

Comparing pattern distributions

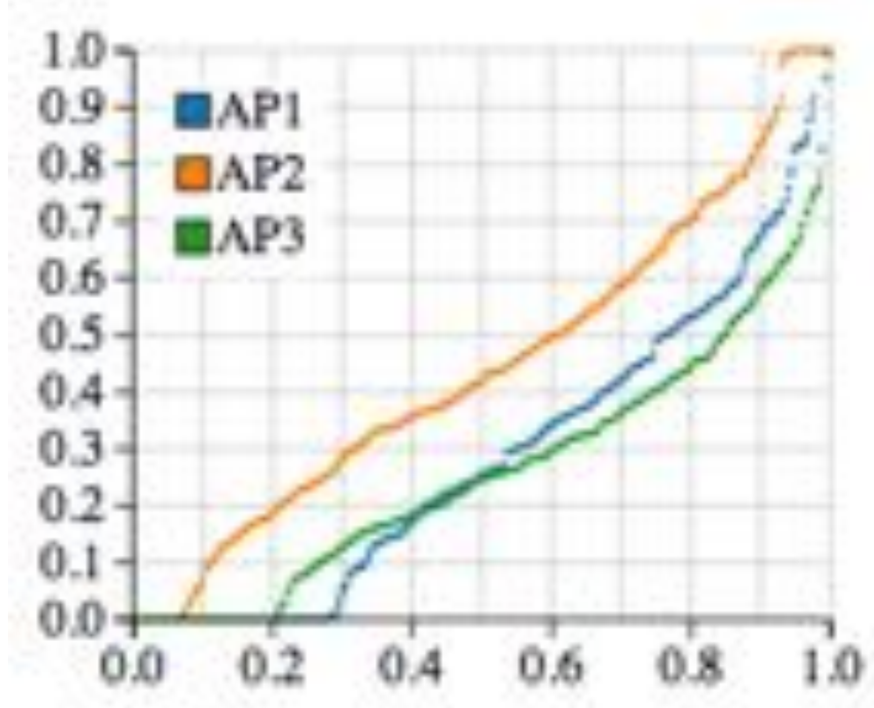
First 30 days

$\mathcal{K}=2$



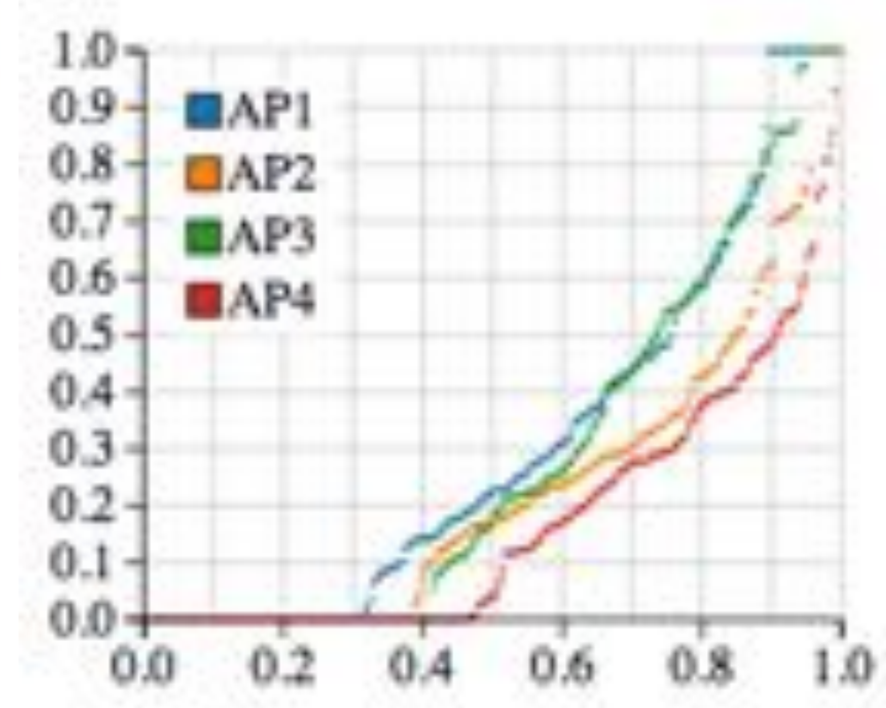
1. Overall
2. Time-partitioned

$\mathcal{K}=3$



1. Overall
2. weaker Overall
3. Time-partitioned

$\mathcal{K}=4$

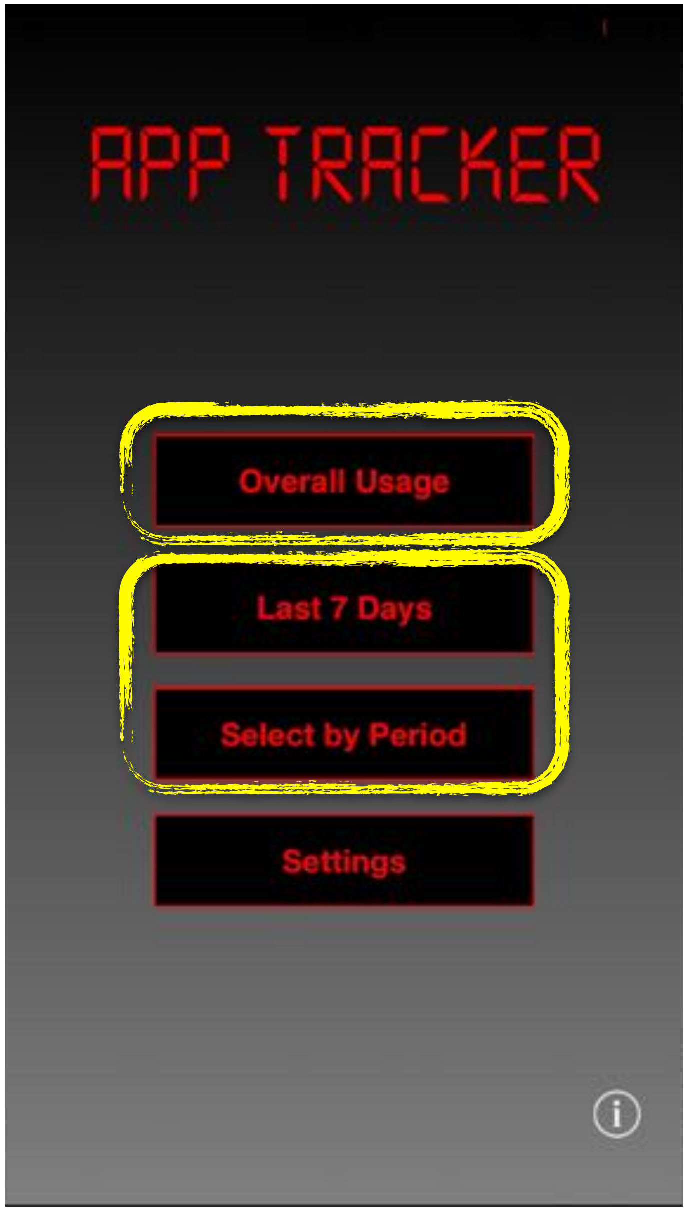


1. mainly `TopApps`
2. `Stats` - `TopApps`
3. Time-partitioned
4. exclusive `TopApps` and `UsageBarChart`

Inform the app redesign

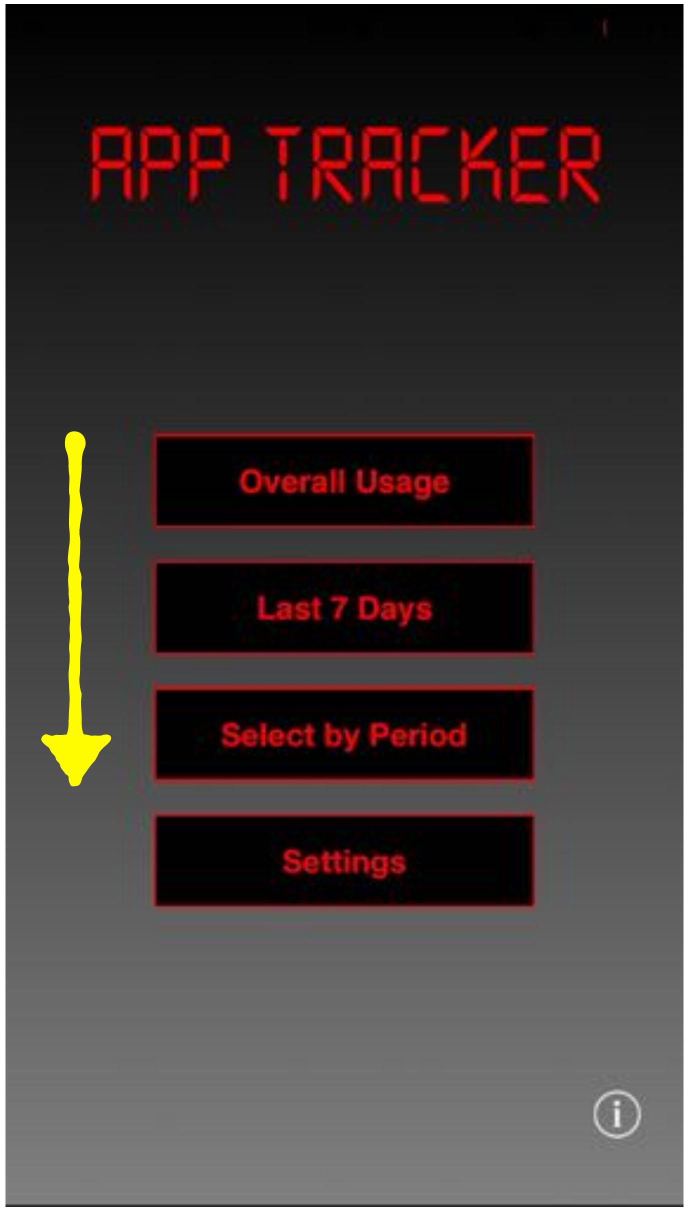
- For $\mathcal{K}$=2 no pattern is significantly dominant, usage is fairly distributed between the two patterns.
- Session length indicative of a more suitable glancing-like view.
- From 3 to 2 main viewing options ?

    1. glancing-like short interactions in a new *Overall Usage* screen
    2. longer interactions in a new ***Select by Period***, including ***Last 7 Days*** and more filtering and querying tools



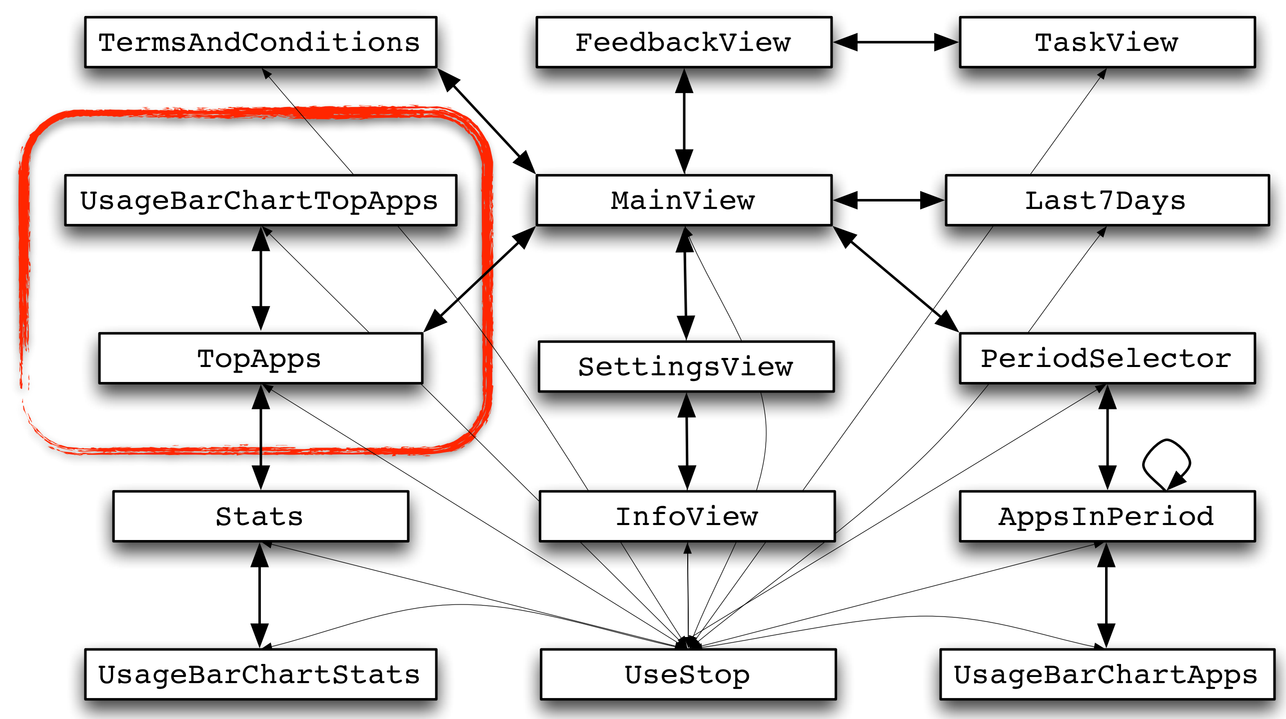
Inform the app redesign

- Are users merely following the suggested paths defined by the interface?
  - For $\mathcal{K}$=3 there is no pattern centred uniquely around ***Select by Period***, but one centred around ***Last 7 Days***, and one around both.
  - For $\mathcal{K}$=4 ***Last 7 Days*** and ***Select by Period*** always go together, the same is true for $\mathcal{K}$=5.



Inform the app redesign

- For $\mathcal{K}$=4 and $\mathcal{K}$=5 we uncover repeated switching between **`TopApps`** and **`UsageBarChartTopApps`**
  - more investigatory than glancing behaviour
  - not just uncovering the menu structure, but finding unexpected behaviours
  - move this loop from ***Overall Usage*** to ***Select by Period***?



Inform the app redesign

- Discovering glancing activity patterns for **widget** extensions on iOS8 and iOS9, or **glances** on the Apple Watch

- Typical glancing patterns for AppTracker are Overall Viewing and `TopApps`-centred patterns

Glances. You can provide users with timely read-only information that they care about with a Glance — a quick and lightweight view of your app.

Conclusion — our contribution

- Populations of users characterised by inferred temporal behaviours rather than user attributes
  - ✦ Inference of Markov models of usage patterns from logged user sessions — activity patterns
  - ✦ Characterisation the activity patterns by probabilistic temporal properties using model checking
- Analysis of a mobile app to inform developers about the actual use and future redesign

Ongoing and future work

- Developing a code environment for the analysis — Blocks
- More apps to analyse and properties to identify:
  - ✦ Activity patterns combined with user attributes (timezone, device type)
  - ✦ Different probabilistic models, e.g., Hierarchical Hidden Markov models
  - ✦ Game app: Hungry Yoshi — for a richer dataset of user traces
  - ✦ Activity tracking/health apps: MatchFIT, Quped — to be released soon

A POPULATION APPROACH TO UBICOMP SYSTEM DESIGN

A Population Approach to Ubicomp System Design is a five year research programme working towards a new science of software structures.

The Populations research programme is funded by an EPSRC Programme Grant (EP/J007617/1). It is a collaboration between the University of Glasgow and the University of Warwick.

Drawing metaphorically from biological concepts of species and evolution, *The Populations Programme* accepts and takes advantage of the scale, variety and dynamism possible in contemporary software. We treat software class as a varied and changing population of software instances in use.

www.softwarepopulations.com

Thank you!

Questions?