

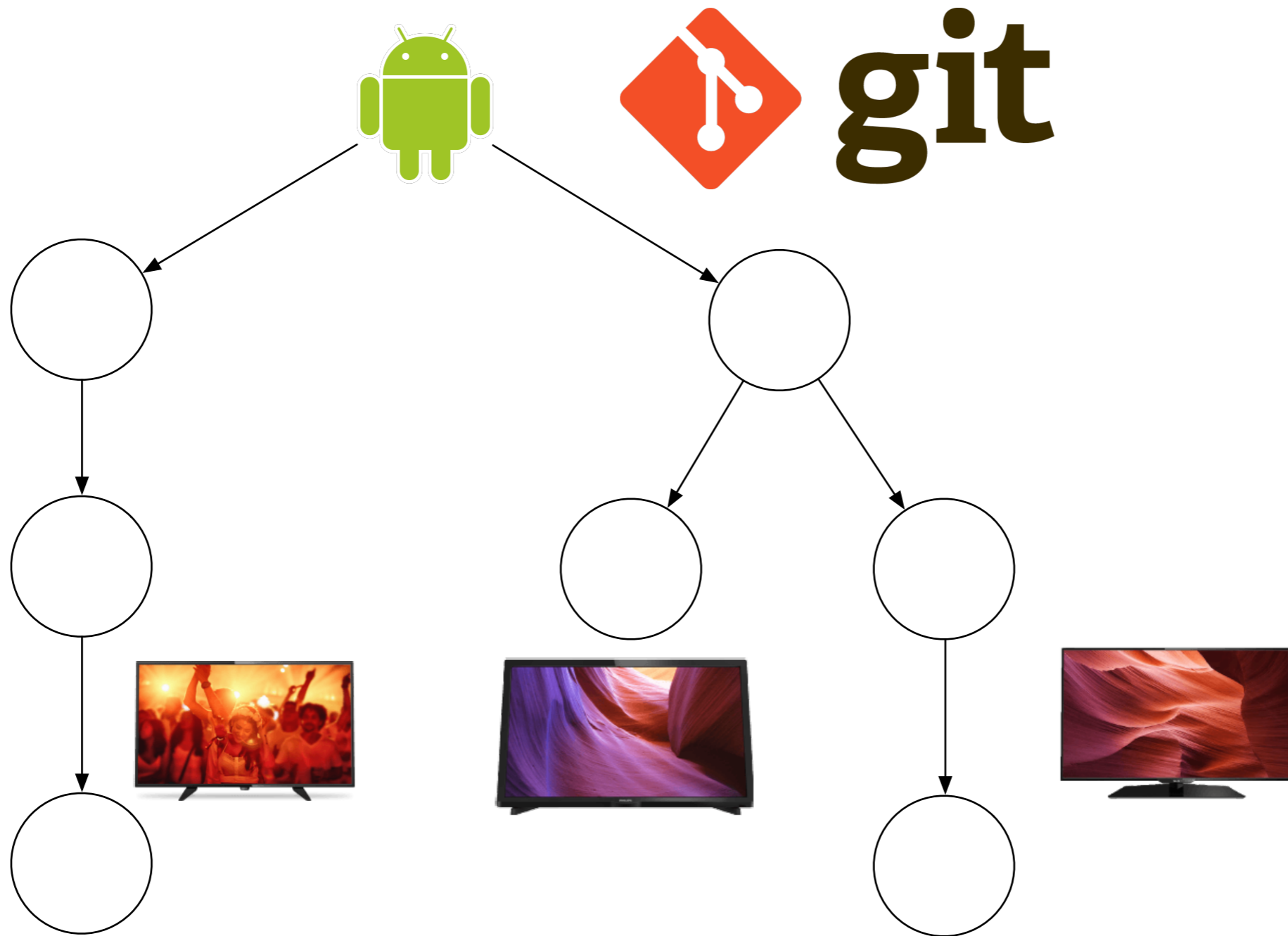


# Mining Repositories for Systematic Edits

Presenter: Reinout Stevens



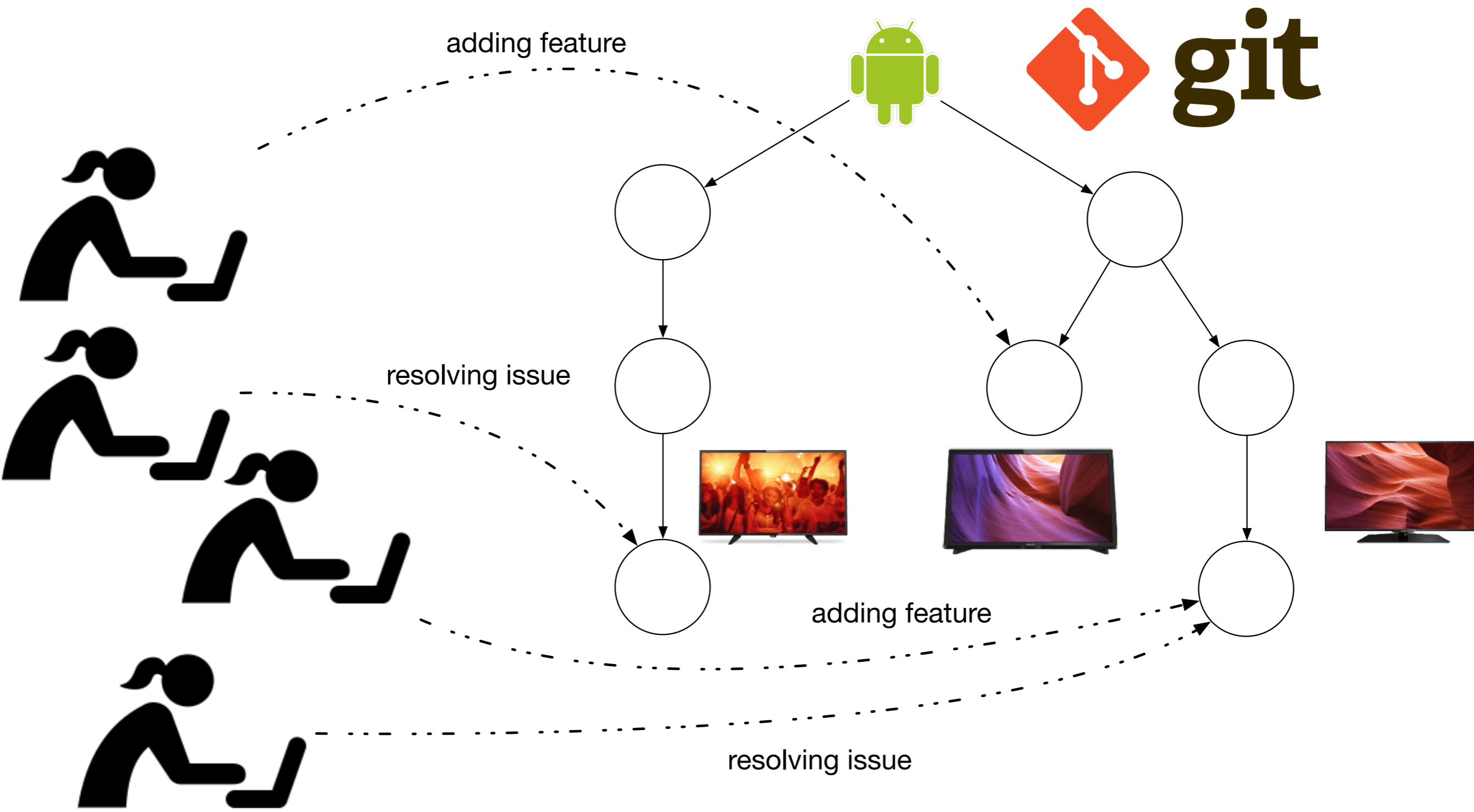
# TPVISION



# TPVISION



git



# Systematic Edits

```
1 1 show_schedule.setOnFocusChangeListener(new OnFocusChangeListener() {
2 2 @Override
3 3 public void onFocusChange(View v, boolean hasFocus) {
4 4 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
4 4 + new Handler().post(new Runnable() {
5 5 + @Override
6 6 + public void run() {
7 7 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
8 8 + });
5 9 }
6 10 });
7 11
8 12 show_schedule.setOnTouchListener(new OnTouchListener() {
9 13 @Override
10 14 public boolean onTouch(View v, MotionEvent event) {
11 14 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
15 15 + new Handler().post(new Runnable() {
16 16 + @Override
17 17 + public void run() {
18 18 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
19 19 + });
12 20 return true;
13 21 }
14 22 });
```

# Systematic Edits

```
1 1 show_schedule.setOnFocusChangeListener(new onFocusChangeListener() {
2 2 @Override
3 3 public void onFocusChange(View v, boolean hasFocus) {
4 4 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
4 4 + new Handler().post(new Runnable() {
5 5 + @Override
6 6 + public void run() {
7 7 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
8 8 + });
5 9 }
6 10 });
7 11
8 12 show_schedule.setOnTouchListener(new onTouchListener() {
9 13 @Override
10 14 public boolean onTouch(View v, MotionEvent event) {
11 11 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
15 15 + new Handler().post(new Runnable() {
16 16 + @Override
17 17 + public void run() {
18 18 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
19 19 + });
12 20 return true;
13 21 }
14 22 });
```

# Systematic Edits

```
1 1 show_schedule.setOnFocusChangeListener(new onFocusChangeListener() {
2 2 @Override
3 3 public void onFocusChange(View v, boolean hasFocus) {
4 4 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
4 4 + new Handler().post(new Runnable() {
5 5 + @Override
6 6 + public void run() {
7 7 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
8 8 + });
5 9 }
6 10 });
7 11
8 12 show_schedule.setOnTouchListener(new onTouchListener() {
9 13 @Override
10 14 public boolean onTouch(View v, MotionEvent event) {
11 11 - optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
15 15 + new Handler().post(new Runnable() {
16 16 + @Override
17 17 + public void run() {
18 18 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
19 19 + });
12 20 return true;
13 21 }
14 22 });
```

# Systematic Edits

```
1 1 show_schedule.setOnFocusChangeListener(new onFocusChangeListener() {
2 2 @Override
3 3 public void onFocusChange(View v, boolean hasFocus) {
4 4 - optionsfocuschange(OPTIONS SHOW SCHEDULE);
5 5 + new Handler().post(new Runnable() {
6 6 + @Override
7 7 + public void run() {
8 8 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
9 9 + });
10 10 });
11 11
12 12 show_schedule.setOnTouchListener(new onTouchListener() {
13 13 @Override
14 14 public boolean onTouch(View v, MotionEvent event) {
15 15 - optionsfocuschange(OPTIONS SHOW SCHEDULE);
16 16 + new Handler().post(new Runnable() {
17 17 + @Override
18 18 + public void run() {
19 19 + optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
20 20 + });
21 21 return true;
22 22 }
23 23 });
```

# Overview of the Approach



```
1 show_schedule.setOnFocusChangeListener(new onFocusChangeListener() {
2     @Override
3     public void onFocusChange(View v, boolean hasFocus) {
4         optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
5     }
6     + new Handler().post(new Runnable() {
7         @Override
8         public void run() {
9             optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
10        }
11    });
12 }
13 }
14 }
15 }
16 }
17 }
18 }
19 }
20 }
21 }
22 }
```

Mine Repository

Compute Changes

```
public <method-name>(View <param1>, <type> <param2>) {
    optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
}
```



```
double <method-name>(View <param1>, <type> <param2>) {
    new Handler().post(new Runnable() {
        @Override
        public void run() {
            optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
        }
    })
}
```

Detect Systematic Edits



# Computing Changes

```
show_schedule.setOnFocusChangeListener(new ... {  
@Override  
public void onFocusChange(View v, boolean hasFocus) {  
    optionsfocuschange(OPTIONS_SHOW_SCHEDULE);  
}  
});
```

```
show_schedule.setOnTouchListener(new ...() {  
@Override  
public boolean onTouch(View v, MotionEvent event) {  
    optionsfocuschange(OPTIONS_SHOW_SCHEDULE);  
    return true;  
}  
});
```

```
show_schedule.setOnFocusChangeListener(new ... {  
@Override  
public void onFocusChange(View v, boolean hasFocus) {  
    new Handler().post(new Runnable() {  
        @Override  
        public void run() {  
            optionsfocuschange(OPTIONS_SHOW_SCHEDULE);  
        }  
    });  
}  
});
```

```
show_schedule.setOnTouchListener(new ...() {  
@Override  
public boolean onTouch(View v, MotionEvent event) {  
    new Handler().post(new Runnable() {  
        @Override  
        public void run() {  
            optionsfocuschange(OPTIONS_SHOW_SCHEDULE);  
        }  
    });  
    return true;  
}  
});
```

Move

Insert

1. insert(:body, onFocusChange, new Handler())
2. insert(:identifier,new Hander(),post())
3. insert(:arguments,post(),new Runnable())
4. insert(:bodyDeclaration,new Runnable(), public void run())
5. ...
6. move(:body, run(), optionsfocuschange())
7. ...

# Frequent Itemset Mining

| Group | Items                 |
|-------|-----------------------|
| 1     | {Bread, Butter}       |
| 2     | {Beer, Milk, Butter}  |
| 3     | {Bread, Milk, Butter} |
| 4     | {Beer, Butter}        |
| 5     | {Bread, Milk, Butter} |

# Frequent Itemset Mining

| Group | Items                 |
|-------|-----------------------|
| 1     | {Bread, Butter}       |
| 2     | {Beer, Milk, Butter}  |
| 3     | {Bread, Milk, Butter} |
| 4     | {Beer, Butter}        |
| 5     | {Bread, Milk, Butter} |



Frequent pattern (support 3)  
{Bread, Butter}

~~Frequent pattern (support 3)  
{Bread}~~

...

# Applied to Changes

| Group | Items                 |
|-------|-----------------------|
| 1     | {Bread, Butter}       |
| 2     | {Beer, Milk, Butter}  |
| 3     | {Bread, Milk, Butter} |
| 4     | {Beer, Butter}        |
| 5     | {Bread, Milk, Butter} |

Q1. How to group changes?

Q2. When are two changes considered equal?

# Grouping Analogy

| Group | Items                 |
|-------|-----------------------|
| 1     | {Bread, Butter}       |
| 2     | {Beer, Milk, Butter}  |
| 3     | {Bread, Beer, Butter} |

| Group | Items                   |
|-------|-------------------------|
| 1     | {Saw, Gloves, Nails}    |
| 2     | {Hammer, Nails, Wood}   |
| 3     | {Nails, Gloves, Hammer} |

| Group | Items                                       |
|-------|---|
| 1     | {Coughing Syrup}                            |
| 2     | {Band Aid, Painkillers,<br>Rubbing Alcohol} |
| 3     | {Band Aid, Painkillers}                     |

# Grouping Analogy

| Group | Items                 |
|-------|-----------------------|
| 1     | {Bread, Butter}       |
| 2     | {Beer, Milk, Butter}  |
| 3     | {Bread, Beer, Butter} |

| Group | Items                   |
|-------|-------------------------|
| 1     | {Saw, Gloves, Nails}    |
| 2     | {Hammer, Nails, Wood}   |
| 3     | {Nails, Gloves, Hammer} |

| Group | Items                                    |
|-------|--|
| 1     | {Coughing Syrup}                         |
| 2     | {Band Aid, Painkillers, Rubbing Alcohol} |
| 3     | {Band Aid, Painkillers}                  |

| Group | Items   |
|-------|---|
| 1     | {Bread, Butter, Saw, Gloves, Nails, Syrup}  |
| 2     | {Beer, Milk, Butter, Hammer, Nails, Wood, Band Aid, Painkillers, Rubbing Alcohol} |
| 3     | {Bread, Milk, Butter, Nails, Gloves, Hammer, Band Aid, Painkillers}               |

# Grouping Changes

| Group           | Changes   |
|-----------------|---|
| onFocusChange() | <pre>insert(:body, onFocusChange, new Handler()) insert(:identifier,new Hander(),post()) insert(:arguments,post(),new Runnable()) insert(:bodyDeclaration,new Runnable(), public void run()) move(:body, run(), optionsfocuschange()) ...</pre> |
| onTouch()       | <pre>insert(:body, onTouch, new Handler()) insert(:identifier,new Hander(),post()) insert(:arguments,post(),new Runnable()) insert(:bodyDeclaration,new Runnable(), public void run()) move(:body, run(), optionsfocuschange()) ...</pre>       |

Method

File

Package

Commit

# Equality Analogy

| Transaction | Items                                    |
|-------------|--|
| 1           | {White Bread, Butter}                    |
| 2           | {Alcohol-Free Beer, Soy Milk, Margarine} |
| 3           | {Brown Bread, Whole Milk, Butter}        |
| 4           | {Trappist, Chocolate Paste}              |
| 5           | {French Bread, Skimmed Milk, Butter}     |



**Frequent pattern (support 3)**  
{Bread, Butter}



```

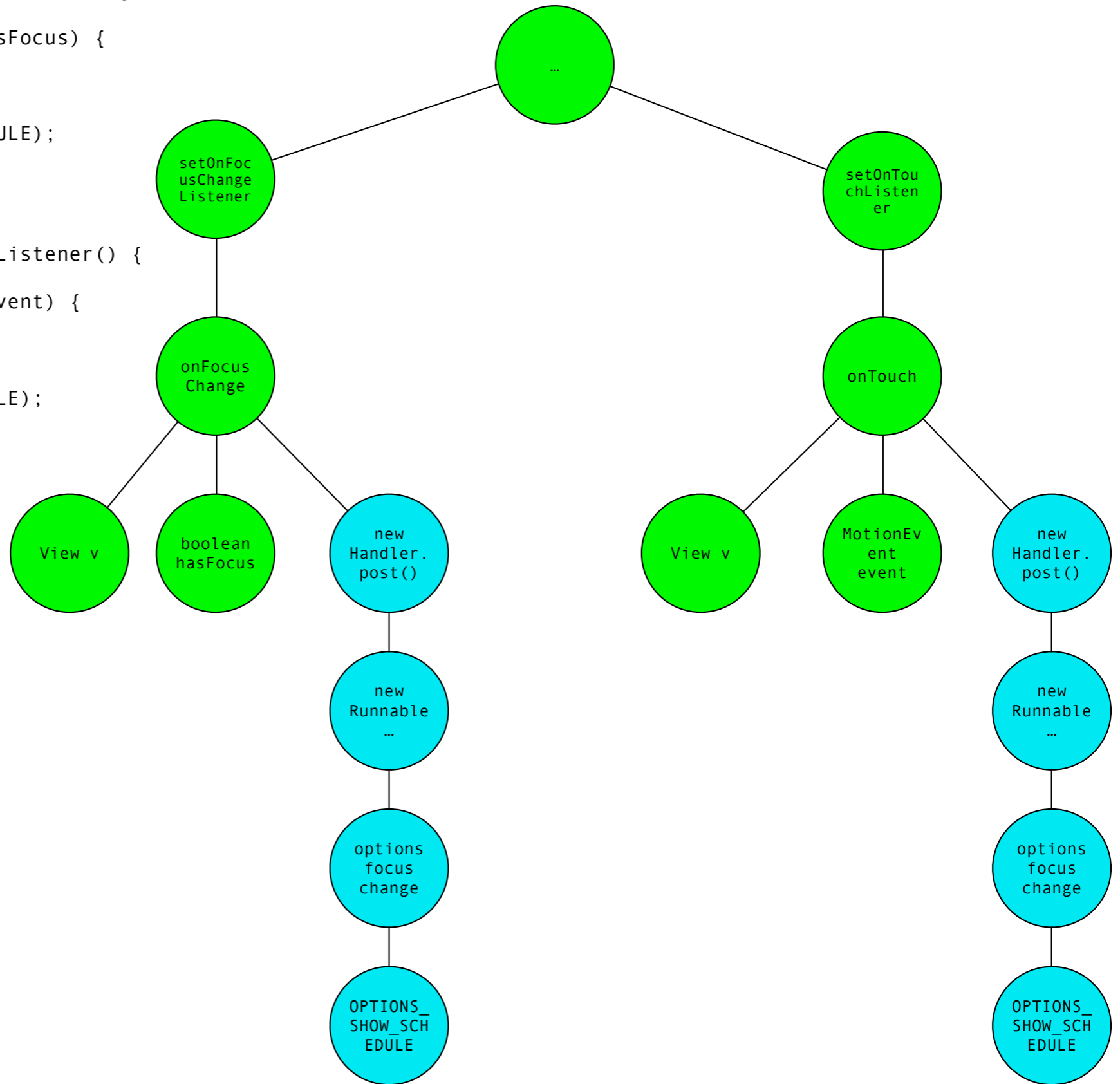
show_schedule.setOnFocusChangeListener(new OnFocusChangeListener() {
@Override
public void onFocusChange(View v, boolean hasFocus) {
    new Handler().post(new Runnable() {
        @Override
        public void run() {
            optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
        }
    });
}
});

```

```

show_schedule.setOnTouchListener(new OnTouchListener() {
@Override
public boolean onTouch(View v, MotionEvent event) {
    new Handler().post(new Runnable() {
        @Override
        public void run() {
            optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
        }
    });
    return true;
}
});

```



 Context

 Modified

# TP Vision Data



~1650 Commits



~300 Classes



Grouped by method

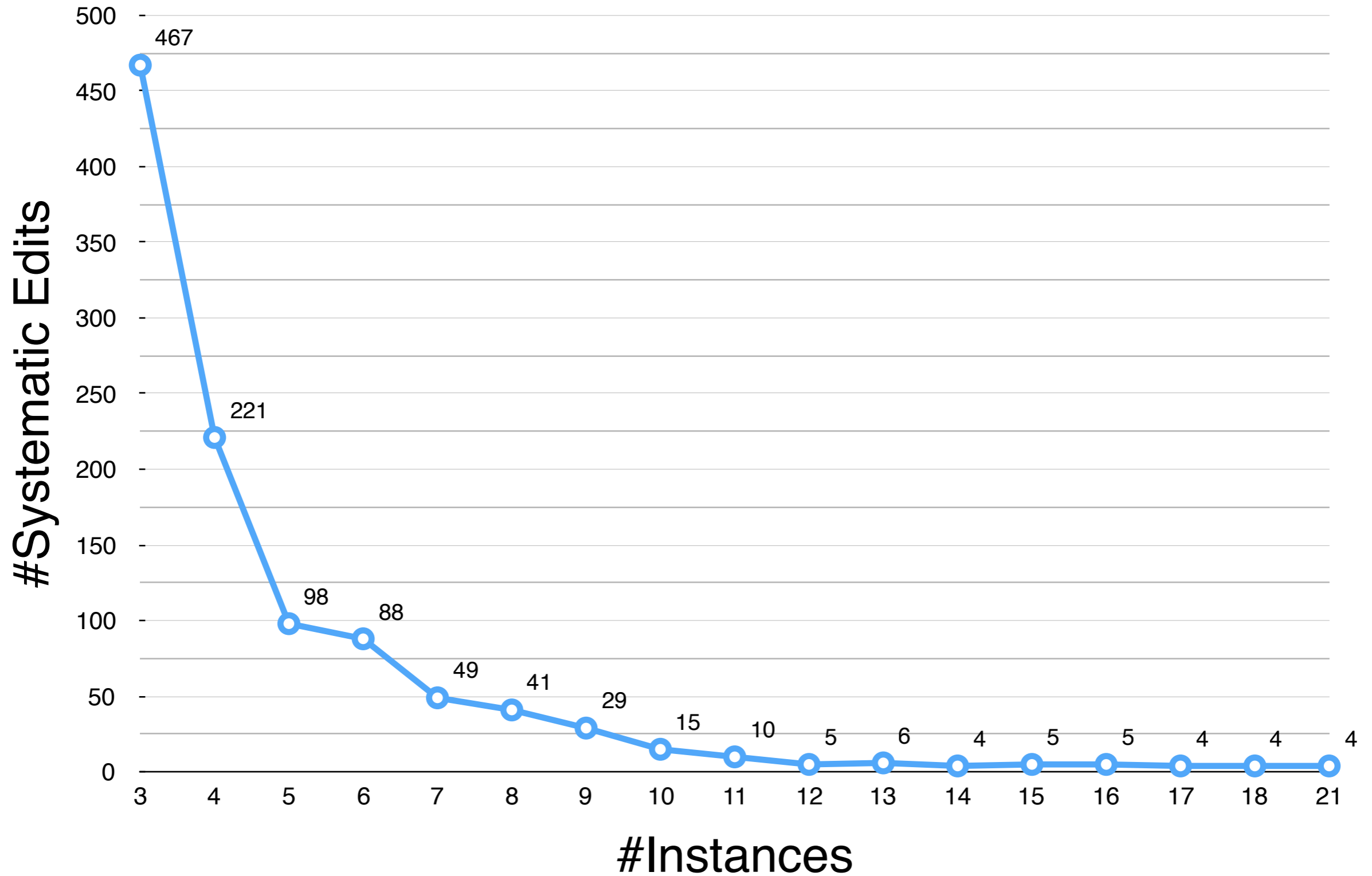


~27000 LOC

=?

Structurally equal

# Results



# Example: Threading

```
show_schedule.setOnFocusChangeListener(new OnFocusChangeListener() {
    @Override
    public void onFocusChange(View v, boolean hasFocus) {
+   new Handler().post(new Runnable() {
+       @Override
+       public void run() {
+           optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
+       });
    }
});

show_schedule.setOnTouchListener(new OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
+   new Handler().post(new Runnable() {
+       @Override
+       public void run() {
+           optionsfocuschange(OPTIONS_SHOW_SCHEDULE);
+       });
    return true;
    }
});
```

# Example: Disable

```
- Log.i(TAG,"programdata dbSize  
>IPEPGClientServiceConstants.MAX_DB_SIZE"+ dbSize);  
  
+//Log.i(TAG,"programdata dbSize  
>IPEPGClientServiceConstants.MAX_DB_SIZE"+ dbSize);  
  
- Log.i(TAG,"programdata dbSize is greater than max limit,  
deleting programs for scid "+ scid);  
  
+ //Log.i(TAG,"programdata dbSize is greater than max limit,  
deleting programs for scid "+ scid);  
  
...
```

# Example: Conversion

```
-     private Tuning  plfTuning;
-     private TuningDvbC plfTuningDvbc ;
-     private TuningDvbT plfTuningDvbt;
-     private TuningDvbSEx plfTuningDvbs;
-     private OnChangeListener listener;
+     private ITuning  plfTuning;
+     private ITuningDvbC plfTuningDvbc;
+     private ITuningDvbT plfTuningDvbt;
+     private ITuningDvbS plfTuningDvbs;
+     private TvMediaPlayer tvMpl;
+     private ITvMediaPlayerContext itvmpIcTtxt;
+     private ITuning.OnChangeListener listener;
```

# Conclusion



Mine Repository for  
Systematic Edits

