

Porting Applications Across Platforms

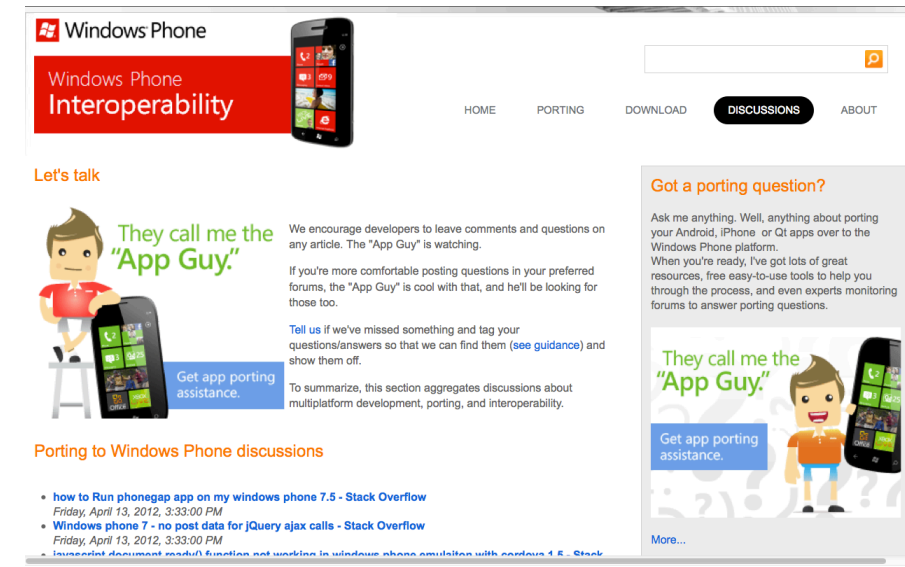
Need to develop applications for each platform



- Can re-use existing code-base during porting
- Porting applications across platforms is challenging
- Have to handle platform-specific code

API Mapping

- A challenging task in porting: Find equivalent functions between platform-specific APIs
- Manual ways exist to find API mappings

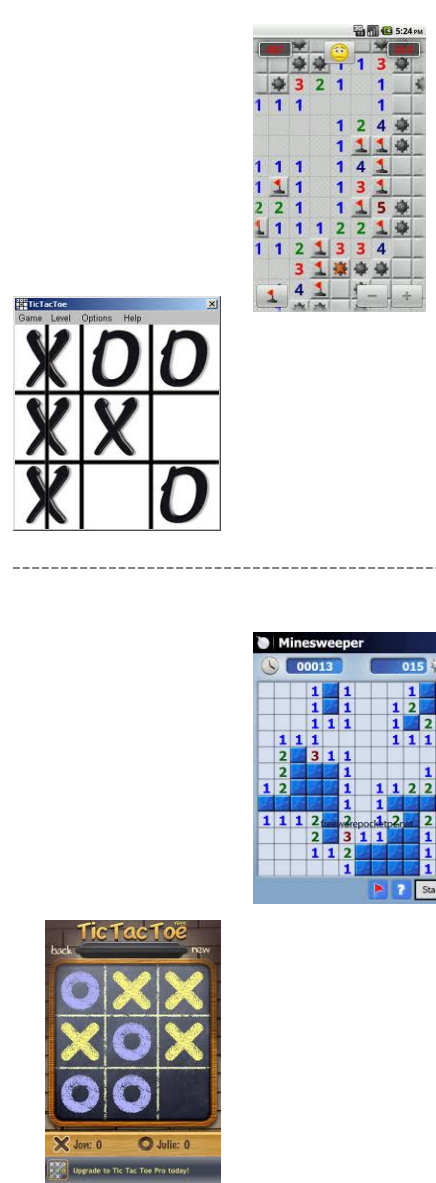


“App Guy”: Provides windows phone porting assistance by giving mappings between APIs

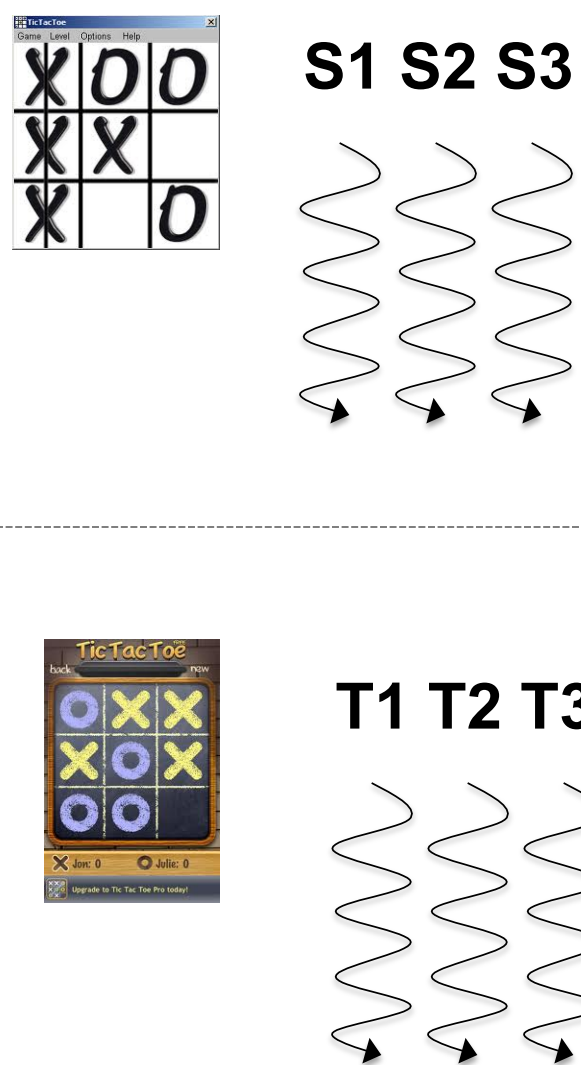
- Proposed an automated approach to find mappings between APIs across platforms
- Developed a prototype tool to find mappings between graphics APIs of JavaME and Android

Our Approach

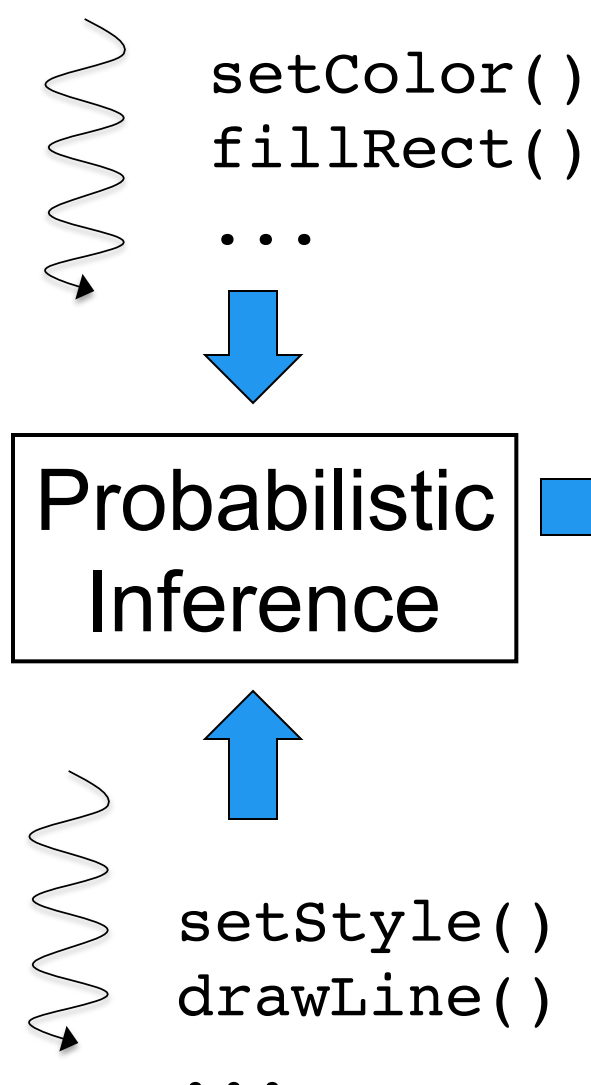
Step 1:
Collect
Apps



Step 2:
Get Execution
Trace pairs



Step 3:
Infer mappings for each trace pair

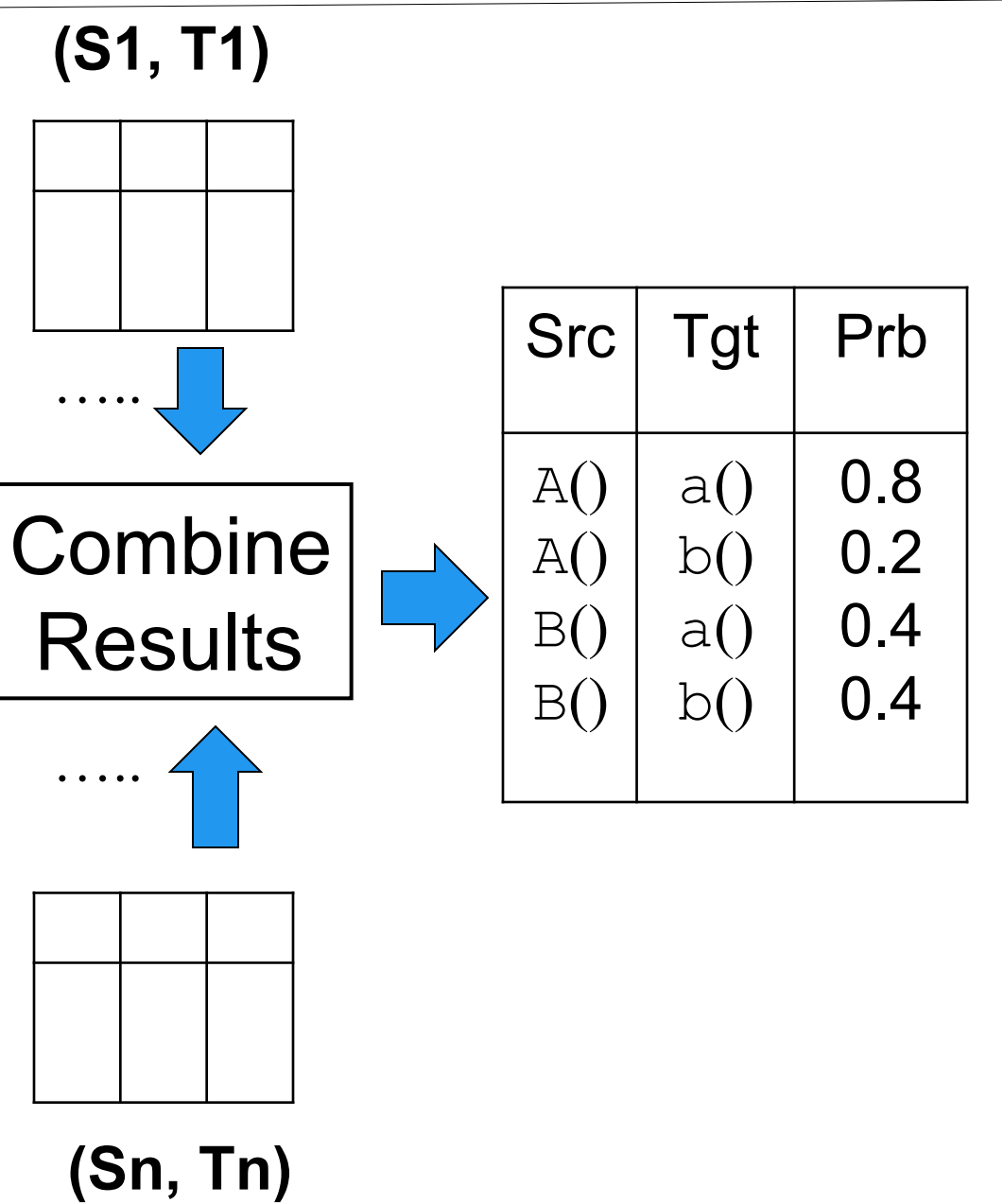


(S1, T1)		
Source	Target	Probability
setColor()	setStyle()	0.9
setColor()	drawLine()	0.1
fillRect()	setStyle()	0.2
fillRect()	drawLine()	0.8

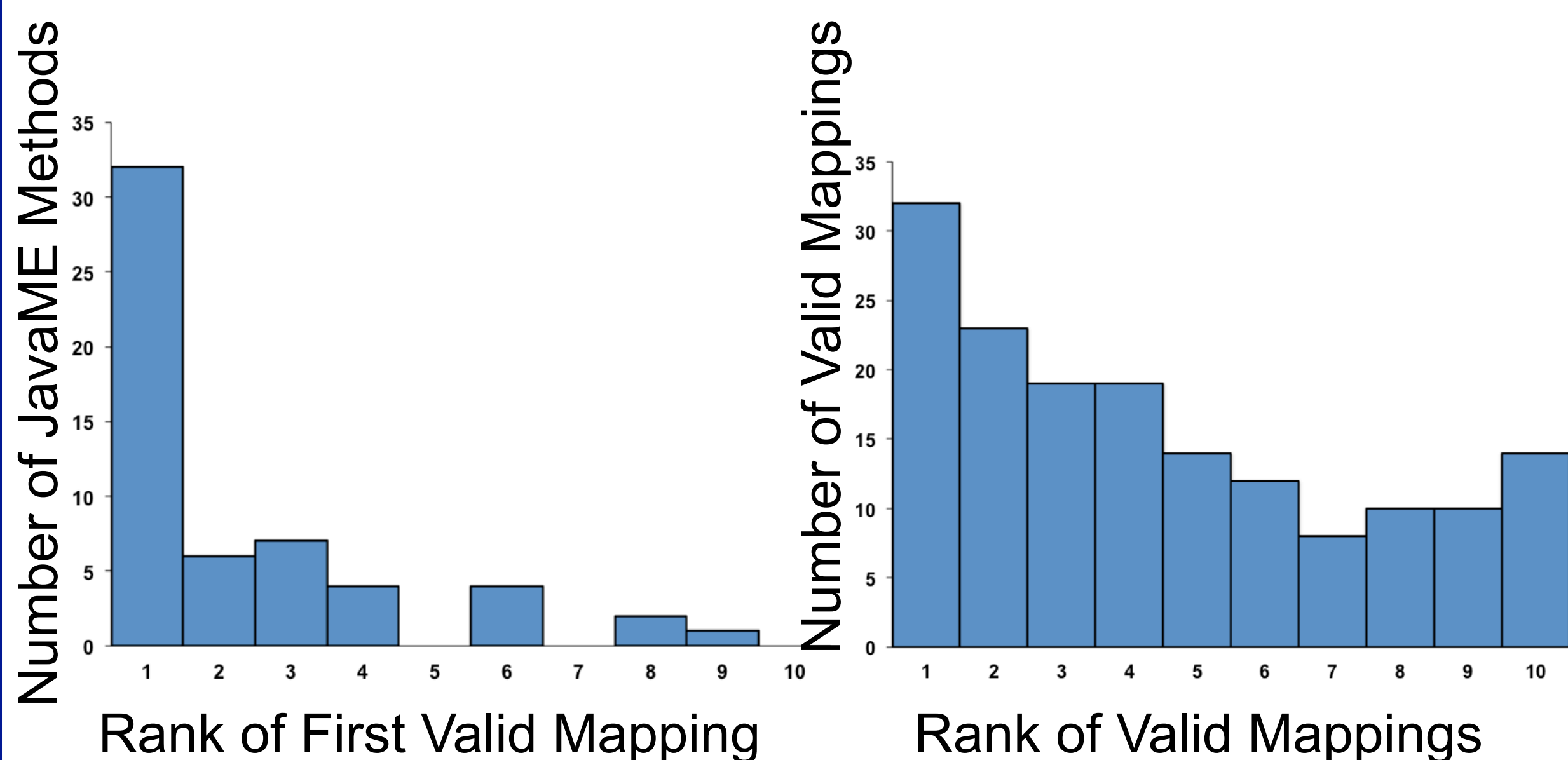
Intuition: Similar methods produce similar visual effects

- Use similarity between attributes of methods
- Attributes: Frequency, position, name, context
- Construct *Factor Graphs*

Step 4:
Combine inferences across
trace pairs



Distribution of Mappings



Summary

Mapping Ex: `Graphics.drawChar()` :
`{ Paint.setColor(); Canvas.drawText() }`

- True mappings found within the list of top 10 for 70% of JavaME methods for graphics APIs
- Proposed new approach to infer mappings between APIs
- Need to handle other issues faced during porting of applications