Lecture 12

Programming Virtual Assistants in Natural Language
Natural Language Programming

- Codex: Help programmers to write (buggy) code?
- Genie: Let consumers script what they want to automate in NL!
  - Beyond queries and single API calls
  - There is a long tail of functions
- Implemented in Genie
  - Monitors
  - Rule-based automation
  - Fine-grain sharing with privacy
Outline

1. Monitors
2. Rule-Based Automation
3. Fine-Grain Sharing with Privacy
   • Remote ThingTalk programs & access control
   • Conformance of access control
   • User study
Monitors

- Get notified when some query satisfies the condition
- Amazon has some hard-coded monitors
  - e.g. Notifications from Amazon deliveries
- Genie allows monitoring of every query
Monitoring is Useful

Repeated action
Query: Is it going to rain tomorrow?
Monitor: Let me know when rain is forecast the next day

Continuous data from IoT
Query: Is there motion on the security camera?
Monitor: Tell me if motion is seen on the security camera

User defined filter
Query: Is the AAPL stock below $100?
Monitor: Tell me when the AAPL stock drops to $100

Scarce resource
Query: Is there a Covid appointment available?
Monitor: Tell me when there is a Covid appointment

What other examples can you come up with?
Implementing Monitors

- **Polling**: The execution engine issues queries at a specified interval
- **Push notifications**: The client gets a push when there is new data
  - Web hooks: API server makes HTTP request to client
    - Efficient, but cloud-to-cloud only
  - Streaming (Web sockets): keep open connection from client server
    - Works from any device, keeping connection might be expensive
- **Monitoring functions are useful, but expensive**
  - Affordable on a personal, always-connected device (smart speaker)
Outline

1. Monitors

2. Rule-Based Automation

3. Fine-Grain Sharing with Privacy
   • Remote ThingTalk programs & access control
   • Conformance of access control
   • User study
IFTTT: Simple Function Composition

- A crowdsourced website; supports If-this-then-that routines with APIs
- Users can create IFTTT routines
  - 250,000 unique recipes
  - 90M activated IFTTT rules
- Simple but powerful paradigms: many existing APIs
- Limitations
  - Proprietary and centralized: user must share credentials
  - GUI user interface: no natural language input
    no formal language representation
- A skill in Alexa, Google Assistant
  - but it must first be configured on the web using GUI
Can We Set Up the Rules by Voice?

• Imagine the scenario:
  • You walk into a bathroom, and it is very humid

![User and Genie interactions]

• Genie supports a superset of IFTTT rules by voice
How to Support Rules by Voice?
Extending Assistant Capabilities

1. Design the conversation domain
2. Create a parser
   - Define the semantics in ThingTalk
   - Write templates
   - Synthesize the data
3. Implement the primitives
4. Implement the compiler to run the program
A Compound Statement in ThingTalk

[when =>]? [get, filter]? => [do]?

“every day at 9am play songs by Taylor Swift”

attimer(time=9:00) =>
@com.spotify.song(), contains(artists, “taylor swift”) => @com.spotify.play(playable=id)

- **When**: when the command is to be executed – *when clause*
- **Get**: a query – *noun*
- **Do**: call an action – *verb*
Another Example

when I post a picture to Instagram with #cat,
post it on Twitter with caption “cat”

monitor @instagram.get_pictures(), contains(hashtags, #cat)
⇒ @twitter.post_picture(url = instagram.url, caption = "cat")

when ⇒ do

• Each primitive is a function call to Thingpedia
• Results can be filtered with simple predicates
  • Equality, comparison, array containment, string predicates
• Parameters passed from one function to the next
More Examples

• **Monitor-action** \( (when \Rightarrow do) \)
  - If the AAPL stock goes to $100, buy 10 shares
  - If it is going to rain tomorrow, turn off the sprinkler
  - Whenever I update my FB profile, update my Twitter profile.

• **Query-action** \( (get \Rightarrow do) \)
  - Get all FB photos posted by Jackie, send them to grandma

• **Monitor-query-action** \( (when \Rightarrow get \Rightarrow do) \)
  - If the AAPL stock goes to $100, get my checking account balance, if it is greater than $1000, buy 1 share.
Everyone’s Needs are Different

**Devices**
“when I use my inhaler, record my GPS location in logfile on Box”

**Location**
“Let my Dad know if I am at the hospital”

**People**
Dr. Smith:
“if Bob’s peak flow-meter drops below 180L/min let me know”

**Environment**
Dr. Smith:
“when the air quality index is above 50 and Bob is running, warn him”
Asthma Use Case in ThingTalk

**WHEN** [FILTERS] → **GET** [FILTERS] → **DO**

**FILTERS:** =, <, >, <=, >=, <>, contains, starts with, ends with

When I use my inhaler, get my location, save them to Dropbox.

If I get taken to a hospital, let my dad know.

When the air quality index is above 500, and I am running, send me an SMS.

If my heart rate is above 130, and I am not running, remind me to take a deep breath.
TakeAways: Monitors & Rules

• Consumers are writing event-driven code!
• Saves much more time than "turn on the light"
• Can perform things otherwise impossible
  (monitoring Covid appointments)
Virtual Assistant: a Flexible Software Architecture

Natural Language → Contextual Semantic Parser → ThingTalk → Thingpedia

Workers can automate their workflow

Quiz: Use cases?

Extensible Constructs

Signatures of public APIs & Implementations of Cloud & IoTs

Also: Corporate Skill Repositories
Outline

1. Monitors
2. Rule-Based Automation
3. Fine-Grain Sharing with Privacy
   • Remote ThingTalk programs & access control
   • Conformance of access control
   • User study
How do People Share Nowadays?

• Share data: photos, files, contacts, ...
  • Facebook
    • Easiest solution but a lot of
  • Email and message

• Share IoT devices and services
  • Username & password?
A Survey on What People Want to Share

- Survey of 200 people from general public (MTurk)
- How comfortable would you be to share
  - With a specific person?
  - With a specific person AND under either of these conditions?
Results of the Survey

Role-Based Permission

- Teenage daughter to use credit card
- Amazon courier to unlock door
- Friends to access cloud drive
- Parent/kid to see security cameras
- 10-year-old kid to use Netflix

Attribute-Based Permission

- With a $20 budget limit
- Photos with their faces in them
- Photos in a specific folder
- Between 7 PM to 9 PM
- Free G or PG rated movies

How to provide such control?
User Study on Access Control

• Survey: 20 scenarios, 200 respondents from MTurk

Willingness to share doubles with attribute-based access control
A New Sharing Approach

• Virtual assistants can naturally help us share
  • They can perform all the digital tasks already
    • On our behalf
    • And on our friends / family’s behalf
  • They can be taught to understand natural language
    • Access control constraints need natural language
A Sharing Work Flow with One User

Dad

“Ask @alice to notify me when her security camera detects motion.”

σ=＠dad, ϵ=SELF: 
monitor @security_camera.event(), has_motion=true ⇒ return

Policy Database

Check
(b)
Save
(e)

σ=＠dad, c=SELF: monitor @security_camera.event(), has_motion=true && @phone.get_gps() {location≠home} ⇒ return

“@dad wants to get notified when any event is detected on your security camera and has motion is equal to true.”

(c) Ask for permission

Alice’s Assistant

Alice

(f) Return detected events

“Notification from monitor security camera: motion detected ...”

(d) Respond

“Only if I’m not home.”

(a) Request
## Sharing with Virtual Assistants

<table>
<thead>
<tr>
<th>Applicability</th>
<th>Convenience</th>
<th>Security</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anything that the assistant can do for the user can be shared</td>
<td>No need to set up access controls ahead of time</td>
<td>The assistant reads to the user exactly what will be executed</td>
<td>The assistant shares precisely what is allowed.</td>
</tr>
<tr>
<td>Once set up, no intervention by the owner is needed</td>
<td></td>
<td>The exact remote program is approved by the user</td>
<td>Theorem prover ensures the conformance of access control</td>
</tr>
</tbody>
</table>
Multiple Inter-Operable Agents

Examples:

• Get me my friends’ pictures of the Halloween party.
• Notify me when my friends are near me!
• Get the linked-in profiles of people at this workshop.
• Schedule an appointment with my boss.

Other examples?
Outline

1. Monitors
2. Rule-Based Automation
3. Fine-Grain Sharing with Privacy
   - Remote ThingTalk programs & access control
   - Conformance of access control
   - User study
Extending Assistant Capabilities

1. Design the conversation domain
2. Create a parser
   - Define the semantics in ThingTalk
   - Write templates
   - Synthesize the data
3. Implement the skills
4. Implement the compiler to run the program
Sharing with Access Control in ThingTalk

Source, Executor: WHEN [FILTERS] → GET [FILTERS] → DO

FILTERS: =, <, >, <>, <=, >=, contains, starts with, ends with

Let Dr. Smith monitor my peak-flow-meter, if it drops below 180L/min.

Let my secretary, whenever I am out of town,
read email messages whose subject is marked urgent.

Let CDC see any of my health records, provided my name is not included.

Let Websites know I am under 13 years old.
Adding Remote Programs to ThingTalk

Ask Alice to notify me when her security camera detects motion

source = SELF , executor = “Alice”: monitor @security_camera.event(), has_motion=true ⇒ return

• source: who asks for the program,
• executor: who executes it
• return: primitive sends result back to the source
Mom or Dad can monitor my security camera only if I am not at home.

- A policy (by the owner)
  - defines a set of allowed executions
- A program (by the requestor)
  - is allowed if it is a subset of the owner’s policy

(source = “Dad” || source = “Mom”),
executor = SELF:
monitor @security_camera.event, @phone.get_gps() {location≠home} ⇒ return
Examples of Access Control

- **Role (Source)**
  - Anybody means publicly available
  - Set predicate (family member)

- **Restriction on operations**
  - Read but not write
  - GPS locations, time of day

- **Arbitrary filters**
  - Rating of movies
  - Below a certain dollar amount …
Satisfiability Modulo Theories (SMT)

- Satisfiability: the problem of determining whether a formula has a model
  - Model: an assignment that makes the formula true

- SAT: Satisfiability of **propositional formulas (and, or, not)**
  - A model is a truth assignment to Boolean variables that makes the formula true
  - SAT solvers: check satisfiability of propositional formulas
    - Decidable, NP-complete

- SMT: Satisfiability modulo theories
  - Satisfiability of first-order formulas with theories:
    - e.g. Arithmetic, arrays, uninterpreted functions, etc.
    - e.g. \( g(a) = c \land f(g(a)) \neq f(c) \lor g(a) = d \land c \neq d \)
  - SMT Solvers: check satisfiability of SMT formulas with respect to theories
SMT Theories for Access Control

• Theories used
  • Set operations (E.g. Allowed source: a family member)
  • Arithmetic (E.g. can turn on the TV between 5-6pm)
  • Uninterpreted functions: functions are treated as black boxes
    (E.g. Allowed function: “read()”)

• For these theories, SMT is decidable, NP-complete
  • SMT solvers usually finish quickly, though exponential in the worst case
  • Access controls are usually trivial to solve
Convert a Program/Policy into an SMT formula

\[
\text{source} = \text{"Bob" : monitor } \tr{instagram.get_pictures()} , \text{contains(hashtags, } \text{#cat)} \Rightarrow \tr{twitter.post_picture(url = \text{instagram.url, caption = } \text{"cat")}
\]

\[
\sigma = \text{"Bob"} \land Y_{1,url} \land Y_{1,hashtags} = F_{\text{instagram.get_pictures}}(r_1) \land \text{mkHashtag(“cat”)} \in Y_{1,hashtags} \\
\land X_{d,url} = Y_{1,url} \land X_{d,caption} = \text{“cat”}
\]
Conformance of Access Controls

Natural Language

Remote Program $p$
Access control $c$

SMT Formula $L[p], L[c]$

Satisfiability Modulo Theories

$c$ describes the set of program executions $p$ conforms to $c$

if $p$ is a subset of $c$

$p \leq c$

$\equiv L[p] \models L[c]$  \hspace{1cm} ($L[p]$ entails $L[c]$)

$\equiv \neg \text{SAT}(L[p] \land \neg L[c])$

If there is no instance of $p$ that does not satisfy $c$

Create the formula.
Use SMT solver for the answer
Owners Can Add Constraints to Requested Programs

The program $p$

\[
\sigma = \text{@dad}, \epsilon = \text{SELF}: \\
\text{monitor } \text{@security\_camera.event()}, \text{has\_motion=}真 \Rightarrow \text{return}
\]

The policy $c$

\[
\sigma = \text{@dad}, \epsilon = \text{SELF}: \text{monitor} \\
\text{@security\_camera.event()}, \text{has\_motion=}真 \&\& \text{@phone.get\_gps()} \{\text{location}\neq \text{home}\} \Rightarrow \text{return}
\]

$p' = p \land c$ is the least restrictive program that satisfies all the constraints of $p$ and $c$ \(\land\) is the “meet” operator

Run $p'$ if $p' \neq \text{null}$. 
Summary

- Convert programs into SMT formulas
- Conformance
  - Set up an SMT expression
  - Use the SMT solver to decide
- If it is not already conforming
  - Add constraints if possible to make it conform
  - By solving another SMT formula
Outline

1. Monitors
2. Rule-Based Automation
3. Fine-Grain Sharing with Privacy
   • Remote ThingTalk programs & access control
   • Conformance of access control
   • User study
How Expressive Is ThingTalk?

- 220 use cases, collected from Mturk
- Workers unfamiliar with the system, just presented with example

85% are expressible in ThingTalk
End-To-End Prototype (an Android App)
How Users React To Our Prototype

User study with 20 users (students & staff)

Do you like the concept?

Do you like the app?
Would you Use the App?

![Bar chart showing the number of users rating the app from 1 to 5. The chart indicates that the majority of users rate the app 4 or 5.](image-url)
Semantic Parsing Accuracy

- Training set: 522,983 synthetic, 3339 paraphrase
- Paraphrase test set: 702
- Real test set: 132
Conclusions

• Let consumers script their workflow
• Add new capabilities:
  • Define ThingTalk
  • Create templates
  • Create compiler/primitives
• Examples
  • Monitors
  • When-get-do event-driven statements
  • Sharing with fine-grain access control: mapped to SMT