Engineering Human-based Services in Hybrid Computing Systems

Hong-Linh Truong
Faculty of Informatics, TU Wien

hong-linh.truong@tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong
@linhsolar
What this lecture is about?

- Not about crowdsourcing here
  - From service engineering perspectives

- Motivating scenarios

- Human service units

- Provisioning and employing human service units – some frameworks
Scenario

Predictive maintenance company

- Offers services for handling IoT Data
- Offers services for big, data analytics
- Offers services for complex problem solving using human experts

IoT Cloud Platform: <<send data>>
Data Analytics Platform: <<analyze data>>
Expert Provisioning Platform: <<predict and solve problems>>
Sensors: <<monitor>>
Chillers: <<control services>>
Chillers: <<control algorithms>>
Chillers: <<control/configure sensors>>
Chillers: <<notify possible problem>>
Chillers: <<predict and solve problems>>

ASE Summer 2018
Integrated systems of software, things and people services

Cloud-based M2M Platform

Sensors → Gateways → Load Balancer → Event Handling Web Service → MOM → NoSQL Big Data

Control, analyze, deploy

Sensor data

Control, Analysis, Configuration

Elasticity Management Platform

Near-Realtime Data Analytics

Offline Data Analytics

Critical situation msg

Smart Communication

Collective Provisioning

Human Interface (Mobile, Mail, Web)

External Service

Human-based Services

Human services

Elasticity control/management msg

task

critical situation msg

critical situation msg/task

forms

task

execute/manage
Hybrid intelligence

Figure 1: Reasoning capabilities for hybrid intelligence.

But how to program human-based services and software-based services together?
Example: some common tasks in data analytics

We should look more domain-specific tasks than typical crowdsourcing tasks (e.g., for data collection)

Domains: IIoT, e.g., predictive maintenance and remote analytics
Human service units in data analytics -- functions

- Evaluating: is the quality of picture good?
- Classifying: is it a man’s or a woman’s picture?
- Detecting: any unidentified object in a picture?
- Labeling: adding location information of a picture
- Cleansing: remove duplicated pictures
- Steering: the quality of picture is bad, should we continue to merge it with others?
- Evaluating results

How to model such functions for human units? E.g., with REST, serverless, or tasks through queue?
HUMAN SERVICE UNITS
Human service units

Human acting as a „service unit“

- Functions
- Non-functional parameters
- Provisioning mechanisms
- Payment models
- Technical interfaces
- Interaction models
Forms of human services

- **Individual Compute Unit (ICU)**
  - An individual is treated like „a processor“ or “functional unit“. A service can wrap human capabilities to support the communication and coordination of tasks

- **Hybrid Compute Unit (Collective) (HCU)**
  - A set of people and software that are initiated and provisioned as a service for solving tasks

- **Services interfaces can be built**

- **Different pricing models and different quality models**
Human service units – provisioning mechanisms (1)

- An infrastructure can be introduced for accessing many ICUs in a crowd
  - Allow people to register their service unit capabilities
  - Facilitate communication, task bidding, retrieval and result delivery
  - Act like a marketplace: multiple providers and multiple consumers
Human service units – provisioning mechanisms (2)

- An “infrastructure-as-a-service“ for ICUs
  - Facilitate communication, task retrieval and result delivery
  - Single ICU as-a-service provider and multiple consumers
MTurk as an ICU provider

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it’s convenient.
1,102,549 HITs available. View them now.

Make Money by working on HITs
HITs - Human Intelligence Tasks - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:
- Can work from home
- Choose your own work hours
- Get paid for doing good work

Get Results from Mechanical Turk Workers
Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. Get started.

As a Mechanical Turk Requester you:
- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you’re satisfied with the results

Find an interesting task
Work
Earn money

Fund your account
Load your tasks
Get results

Find HITs Now

or learn more about being a Worker

FAQ | Contact Us | Careers at Mechanical Turk | Developers | Press | Policies | State Licensing | Blog | Service Health Dashboard
©2005-2016 Amazon.com, Inc. or its Affiliates

ASE Summer 2018 14
Human service units – provisioning mechanisms (3)

- An „infrastructure-as-a-service“ for HCUs
  - Facilitate communication, task retrieval and result delivery
  - Single HCU as-a-service provider and multiple consumers
Human service units – technical interfaces (1)

![Diagram showing relationships between human service units, REST Web Service, analytics activity, and people.](image)
Human service units – technical interfaces (2)

Web Service (REST)

App/Web

Analytics Activity

Email/SMS/Instant Messaging

Conversational Services

Email/SMS/Instant Messaging

People

Analytics Activity

People
Human service units – interaction model

- Analytics Activity
  - Task queue
  - Scheduler
    - pull
    - ICU/HCU

- Analytics Activity
  - Task queue
  - Scheduler
    - push
    - ICU/HCU

- Analytics Activity
  - direct
  - ICU/HCU
Conversational Features

- Analytics Activity
- Task queue
- Scheduler
- Conversation Service
- ICU/HCU

---

ASE Summer 2018 19
Which are important considerations when interpreting non-functional properties for human service units?
Incorporating human units into complex processes

- How to provision and employ human compute units?
- How to select human units?
- Where to place human units in data analytics and why?
- How to monitor and test human units in data analytics?
Selecting human units

- Do not select at all
  - Let human units bid the tasks
    - E.g., in crowdsourcing platforms

- Static/fix mapping
  - E.g., using static information for human-task mapping

- Simple selection techniques
  - Using the requirement of the task to find the suitable human units based on their capabilities

- Complex selection techniques
  - Utilizing complex dependency graphs to find suitable human units
Placement techniques for human units

- Usually at design time the developer/designer decides
  - Where to put human units
  - Where some triggers should be put in order to invoke human units if needed

- At runtime
  - Find suitable human units
  - Invoke human units

- Placement of human units
  - Application-specific
  - Needs automatic algorithms and supporting tools
PROVISIONING AND EMPLOYING HUMAN SERVICE UNITS-- SOME FRAMEWORKS
Approaches

- Software perform task routing and management
- Software perform the work and invoke human only needed
- Humans and software working together
Qurk system architecture (1)

SELECT c.name
FROM celeb c JOIN photos p
ON samePerson(c.img,p.img)
AND POSSIBLY gender(c.img) = gender(p.img)
AND POSSIBLY hairColor(c.img) = hairColor(p.img)
AND POSSIBLY skinColor(c.img) = skinColor(p.img)

TASK gender(field) TYPE Generative:
Prompt: "<table><tr> 
<td><img src='%s'> </td>
<td>What is this person’s gender? </td>
</table>", tuple[field]
Response: Radio("Gender",
["Male","Female",UNKNOWN])
Combiner: MajorityVote

Qurk system architecture (2)

Jabberwocky approach (1)

Figure 1: Overview of Jabberwocky

Source: Salman Ahmad, Alexis Battle, Zahan Malkani, Sepandar D. Kamvar: The jabberwocky programming environment for structured social computing. UIST 2011: 53-64
Jabberwocky approach (2)

```plaintext
1  map :name => :extract_disease_facts do |key,
    value|
2     facts = RiskExtractor.extract (value)
3
4     for fact in facts do
5       emit (fact["disease"], fact["risk_factor"])
6     end
7  end
8
9  reduce :name => :summarize do |key, values|
10     task = SummarizeFacts.prepare
11        :task_name => "Summarize disease risks:
12            #{key}"
13     task.facts = values
14
15     task.ask do |answer|
16       emit (key, answer)
17     end
18  end
```

Source: Salman Ahmad, Alexis Battle, Zahan Malkani, Sepandar D. Kamvar: The jabberwocky programming environment for structured social computing. UIST 2011: 53-64
import edu.umass.cs.automan.adapters.MTurk_

object SimpleProgram extends App {
  val a' = MTurkAdapter { mt =>
    mt.access_key_id = "XXXX"
    mt.secret_access_key = "XXXX"
  }

  def which_one() = a'.RadioButtonQuestion { q =>
    q.budget = 8.00
    q.text = "Which one of these does not belong?"
    q.options = List(
      a'.Option('oscar', "Oscar the Grouch"),
      a'.Option('kermit', "Kermit the Frog"),
      a'.Option('spongebob', "Spongebob Squarepants"),
      a'.Option('cookie', "Cookie Monster"),
      a'.Option('count', "The Count")
    )
  }

  println("The answer is " + which_one())
}

SW4H approach (2)

- Similar concepts in collaborative working environments but integrated into workflows
- Do not discuss about where and how to select human units

Karastoyanova, Dimka; Dentsas, Dimitrios; Schumm, David; Sonntag, Mirko; Sun, Lina; Vukojevic, Karolina: Service-based Integration of Human Users in Workflow-driven Scientific Experiments. In: Proceedings of the 8th IEEE International Conference on eScience (eScience 2012)
Hybrid compute unit (HCU): a set of service units includes software-based services, human-based services and things-based services that can be provisioned, deployed and utilized as a collective on-demand based on different quality, pricing and incentive models.
VIECOM Highlights: Virtualizing Communication

- Extensible architecture
- Adapters for: email, Dropbox, REST, Android
- Open source and documentation:
  - https://github.com/tuwiendsg/SmartCom


American Society for Engineering Education

VIECOM RAHYMS

https://github.com/tuwiendsg/RAHYMS
Muhammad Z. C. Candra, Hong Linh Truong, Schahram Dustdar:
Provisioning Quality-Aware Social Compute Units in the Cloud. ICSOC 2013: 313-327
IoT & Analytics for Predictive Maintenance Approach (INA4PM)

Predictive maintenance: incidents associated with equipment to be monitored and incidents associated with the big data systems

Where and when human units should be used?

How to invoke humans?
Report incidents and generate human tasks

Alexa/Duplex

Voice (human) ← → Voice (machine) → Task (machine)

Tasks ←→ Voice (machine) ←→ Voice (human)

When a task should be done by a bot or human?
Eloquent: AI + Human Tasks

- https://www.eloquent.ai/
- Combine AI with humans for “task-oriented dialog AI”
- Which domains would it be good for?

Figure source: https://www.eloquent.ai/elk.html
Exercises

- Read mentioned papers
- Analyze pros and cons of existing frameworks for data analytics
- Survey existing algorithms for matching human units to data analytics tasks
- Examine requirements for locating places for human units and implement some algorithms
- Examine monitoring techniques for cloud of human compute units
Thanks for your attention

Hong-Linh Truong
Faculty of Informatics, TU Wien
hong-linh.truong@dsg.tuwien.ac.at
http://www.infosys.tuwien.ac.at/staff/truong
@linhsolar