Engineering Applications for the Internet of Things

Pradeep K. Murukannaiah

Department of Computer Science
North Carolina State University

Pradeep K. Murukannaiah: Short Bio

Education

- PhD candidate in Computer Science, NCSU — 2010–Present
- MS in Computer Science, NCSU — 2011
- BE in Information Science and Engineering, UVCE — 2005

Work Experience

- Intern, Google — Summer 2011
- Intern, Duke University — Summer 2009
- Software Engineer, Alcatel-Lucent — 2005–2008

Professional Service

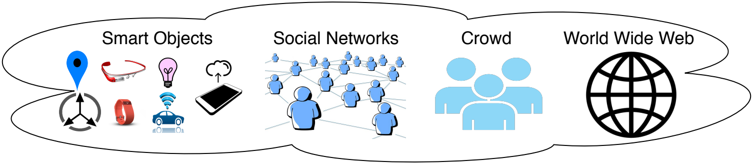
- Information Director, ACM ToIT — 2013–Present
- Reviewer: EMSE, ACM TIST, ACM TSC, IEEE IC
- PC: IEEE MS, AAAI, Social Computing@AAMAS

Outline

Internet of Things (IoT): The next BIG challenge

Engineer applications to deliver a **personalized**, **context-specific**, and **privacy-preserving** experience on the *decentralized* IoT infrastructure

Infrastructure



Potential Applications

- Smart Homes: Automation, health, safety, entertainment
- Smart Cities: Energy, environment, transportation, defense

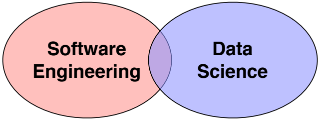
Engineering IoT Applications: Research Challenges

Software Engineering

- Formulate high-level abstractions to represent raw data, e.g., *context*, *trust*, and *norms*
- Incorporate those abstractions into software artifacts, e.g., requirements and design

Data Science

- Map raw sensor data to high-level abstractions of interest via *machine learning*
- Augment machine learning with *human computation* and *collective intelligence*



Software Engineering	Data Science
AAMAS 2014 **	IC 2012 **
RE 2015 **	RecSys 2014
TOSEM 2015 **	AI Mag 2015
	ICSOC 2015
	AAAI 2016
** Lead Author	WWW 2016 **

Empirical Methods

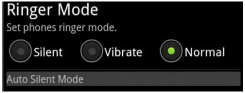
Four **developer** studies
Two **end-user** studies
Two **social media** studies
One **crowdsourcing** study

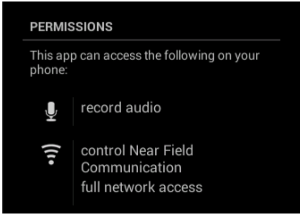
Outline

The Disconnect between Data and Intent

Intelligent Ringer: An example IoT application

Automatically sets ringer modes for incoming calls





Why does Intelligent Ringer consume NFC, audio, and Facebook data?

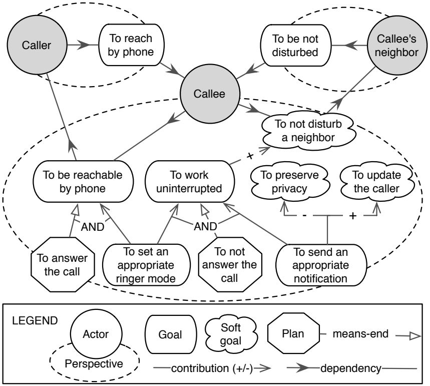
Modeling Intent

Tropos: An agent-oriented methodology

- Metamodel: Actor, goal, plan, belief, resource, and dependency
- Spans requirements, design, specification, and implementation phases

Intents and data are still disconnected

A Tropos model of Intelligent Ringer



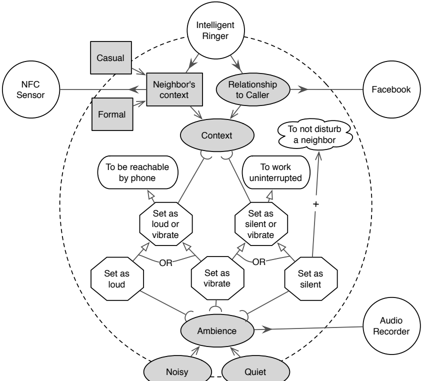
Bridging Data and Intent

Xipho: An extension of Tropos

- Provides systematic steps to map *agent capabilities* to *data abstractions*
- Yields a specification grounded in data, e.g., $Relationship = ?R_1 \wedge Neighbor\text{'}s\ context = Casual \wedge Ambience = Noisy \rightarrow Set\ as\ loud$

Intents and data are now connected

A Xipho model of Intelligent Ringer



Xipho: Empirical Evaluation

What benefits does Xipho offer compared to Tropos?

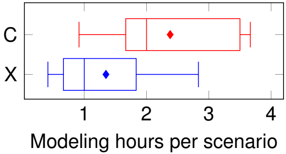
A developer study

- 46 students from a graduate-level computer science course modeled two applications each and comprehended a third
- Application requirements were specified as three–four use cases

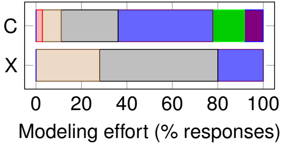
Phase	Task	Group 1	Group 2	Group 3
1	Practice	Alarm	Motivator	Reminder
2	Modeling	Motivator	Reminder	Alarm
3	Verification	Reminder	Alarm	Motivator

Xipho Developer Study: Results

- Xipho significantly reduces **time** and **effort** required to model an application compared to Tropos



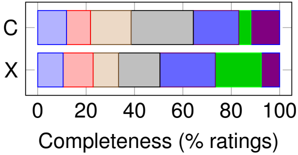
$\mu_X < \mu_C \; (p = 0.046^*)$



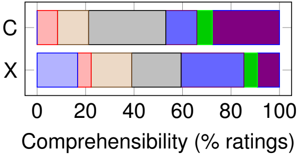
$\widetilde{x}_X < \widetilde{x}_C \; (p = 0.01^{**})$

Xipho Developer Study: Results

- Xipho models of an application are significantly easier to **comprehend** than Tropos models of the application



$\widetilde{x}_X > \widetilde{x}_C \ (p = 0.206)$

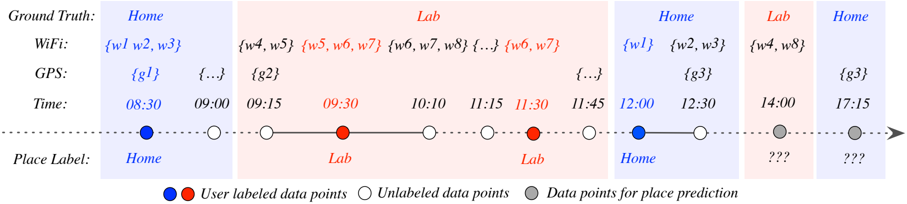


$\widetilde{x}_X > \widetilde{x}_C \ (p = 0.029^*)$

Outline

Characteristics of IoT Data

- **Multimodal** and **sporadic**: Originating from multiple sensors
- **Sparse**: Sensing consumes energy



Stream of intermittent sensor data and place labels

Machine Learning from IoT Data

Unsupervised approaches

- Often require frequent sensing
- Do not capture subjective nuances

Supervised approaches

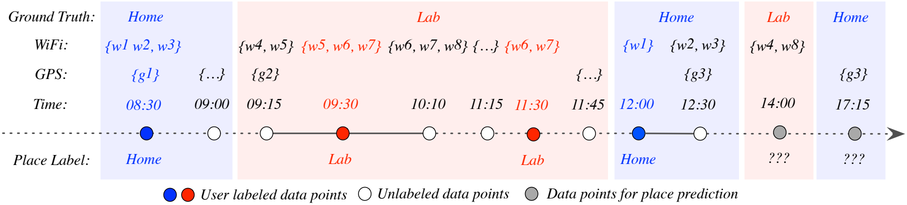
- Require several labels for each place to perform well

Platys: A middle ground solution

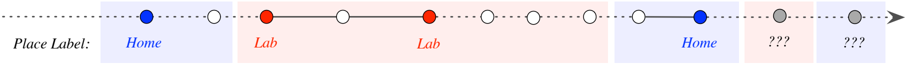
- Active learning seeks to reduce labeling effort
- Semi-supervised learning deals with infrequent sensor data

Active Learning: Intuition

- **Uncertainty sampling:** Ask the user about doubtful predictions



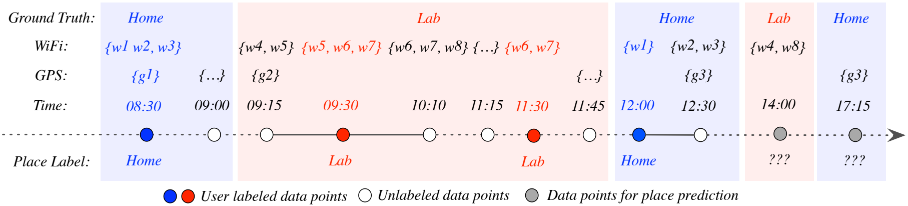
Stream of intermittent sensor data and place labels



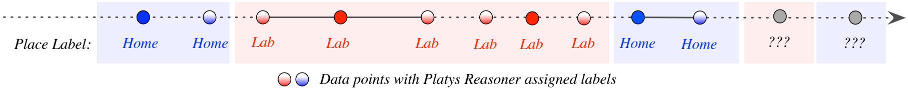
Active learning enables intelligent place labeling

Semi-Supervised Learning: Intuition

- **Self training:** Learn from confident predictions

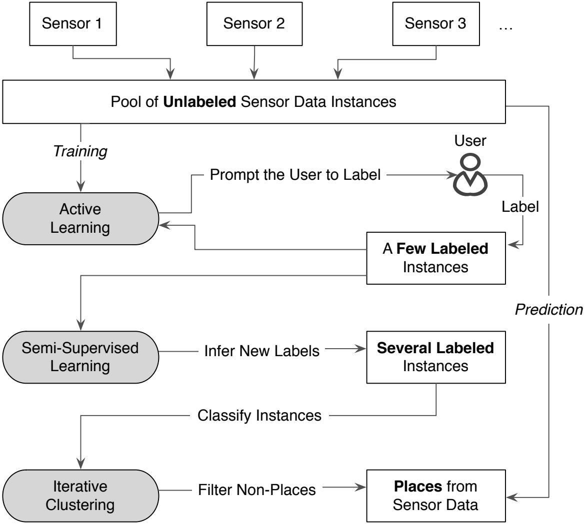


Stream of intermittent sensor data and place labels



Semi-supervised learning exploits latent structure

Platys: An ML Pipeline for IoT



Platys: Empirical Evaluation

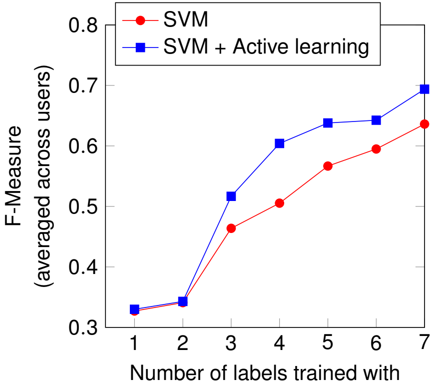
What benefits does Platys offer over traditional supervised and unsupervised ML approaches?

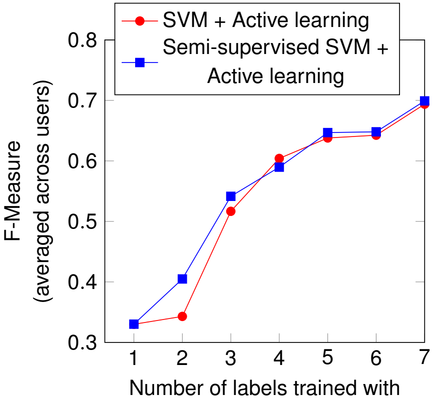
An end-user study

- 10 users employed an Android phone installed with Platys middleware as their primary phone for 10 weeks
- Platys middleware collected sensor readings
  - GPS, WiFi, Bluetooth sensor readings, and Google POI data
- Each user labeled places of his interest multiple times
- Compare Platys to two unsupervised and two supervised approaches after data collection

Platys User Study: Results

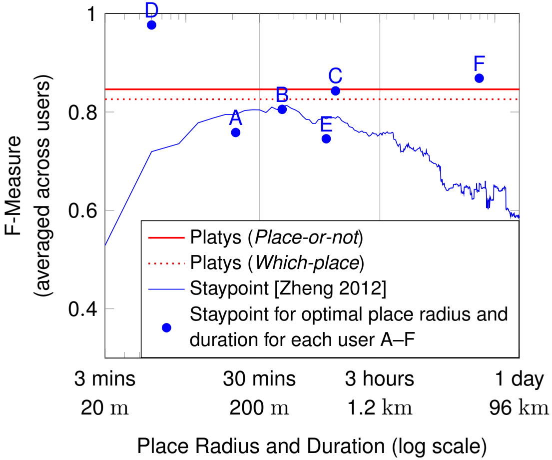
- Platys achieves better F-measure than two traditional classifiers
- Both active learning and semi-supervised learning add value





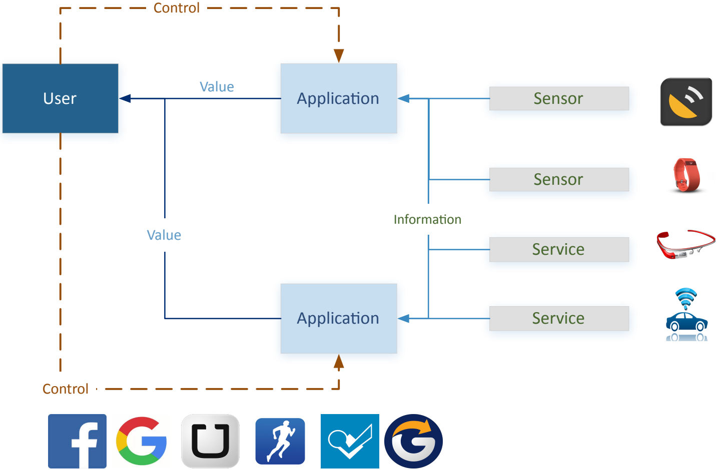
Platys User Study: Results

- Platys achieves better F-measure than two unsupervised approaches
- No single choice of place parameter values is optimal for each user



Outline

IoT Applications: Current Setting



Current Setting: Shortcomings

Potential for privacy loss

- Users' personal data stored by several third-parties

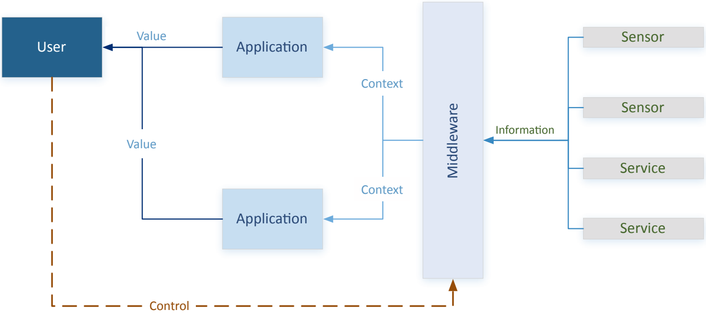
Fragmented user experience

- Each application may interpret users' data differently

Difficult to deliver a personalized experience

- Applications often rely on data aggregated from multiple users

IoT Applications: Proposed Setting



- The *middleware* maps raw sensor data into high-level *contexts* and shares contexts with applications, respecting user's privacy policies

Benefits of the Middleware Setting

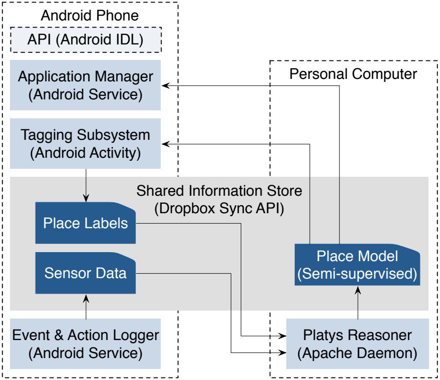
Potential benefits to an end user

- Uniform information architecture
  - Enhances learnability
- Simplified interaction design
  - Enhances ease of use
- Middleware-based service design
  - Preserves privacy

Potential benefits to a developer

- Separation of concerns
  - Simplifies development process
  - Yields modular implementations

Implementation



Platys Middleware: Empirical Evaluation

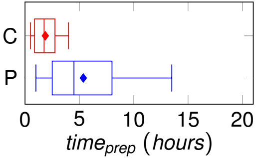
What benefits does Platys middleware offer over traditional setting to an IoT application developer?

A developer study

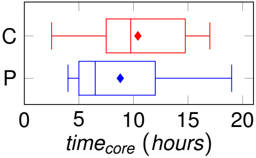
- 46 students from a graduate-level computer science course implemented the Intelligent Ringer application
- Functional requirements specified as three–four use cases
- Usability requirement specified as broad-brush guidelines

Platys Developer Study: Results

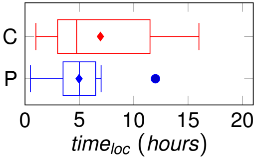
- Although developers employing Platys middleware spend more **time** in preparation, the extra effort pays off in later stages of implementation



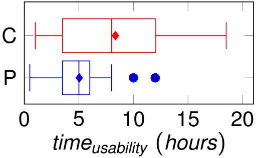
$\mu_P \neq \mu_C$ ** $(p = 0.001)$



$\mu_X \neq \mu_C$ $(p = 0.311)$



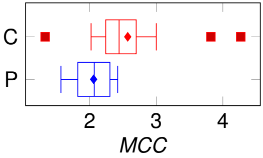
$\mu_P < \mu_C$ * $(p = 0.081)$



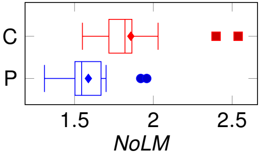
$\mu_P < \mu_C$ ** $(p = 0.021)$

Platys Developer Study: Results

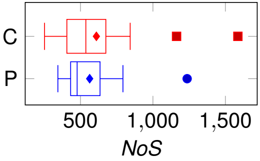
- Applications implemented on the middleware setting are more **modular** than those implemented on the traditional setting



$\mu_P < \mu_C$ **
$(p = 0.001)$



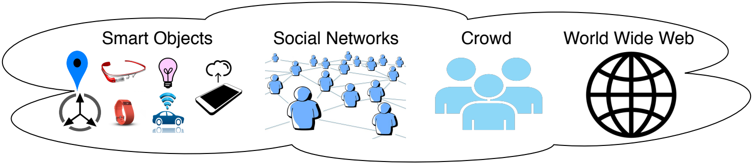
$\mu_P < \mu_C$ **
$(p = 0.001)$



$\mu_P < \mu_C$
$(p = 0.272)$

Outline

Human Intelligence



- Humans are an integral part of the IoT
- No substitute for **human intelligence**, especially, for **creative** tasks

Can we systematically augment data analytics with human intelligence to accomplish creative tasks?

Resolve Goal Conflicts during Requirements Engineering

Intuition: Belief-based goal processing

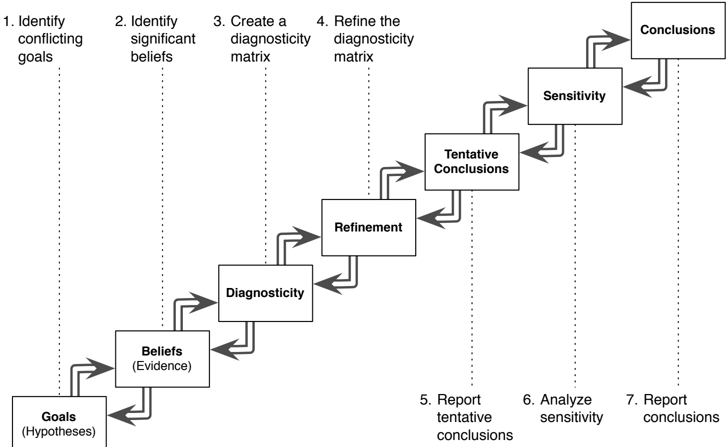
- In order to activate, promote, drop, or suspend a goal, one has to provide or modify the appropriate beliefs [Castelfranchi and Paglieri, 2007]
- Elicit beliefs that support or oppose goals to resolve the conflict

Method: Combine data analytics and human intelligence

- **Analytic method**: Analysis of Competing Hypotheses (ACH)
- **Human intelligence**: Argumentation

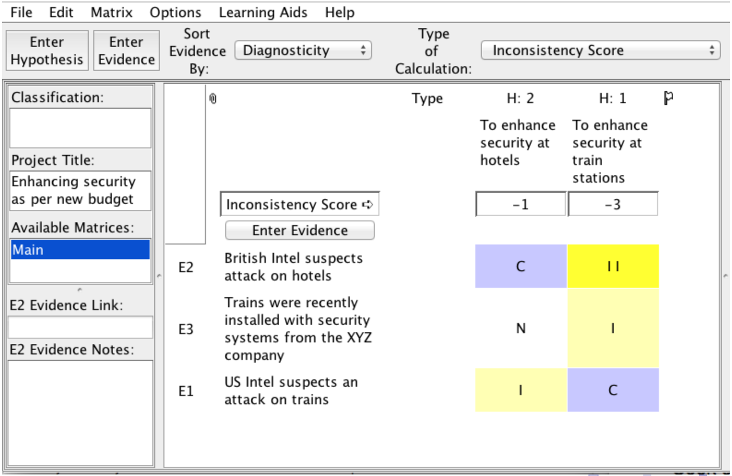
Analysis of Competing Hypotheses (ACH)

ACH is a tool to aid judgement on issues requiring careful weighing of alternatives [Heuer and Pherson, 2014]



ACH Solution to a Case Study

- The PARC ACH tool can be used to construct the **diagnosticity matrix** and analyze **sensitivity** of conclusions



Argumentation-Based ACH (Arg-ACH)

Operates under the rubric of ACH, but with the following exceptions

- Construct an *argument graph* instead of a diagnosticity matrix
- Employ a set of domain-specific and domain-independent *argumentation schemes* [Walton et al., 2008] as critical thinking aids
- Assign *belief* (B), *disbelief* (D), and *uncertainty* (U) scores to each argument, such that $B + D + U = 1$ [Jøsang et al., 2001]

Argumentation Schemes

An argumentation scheme is an inferential structure: a reusable pattern of reasoning. For example, consider the scheme below

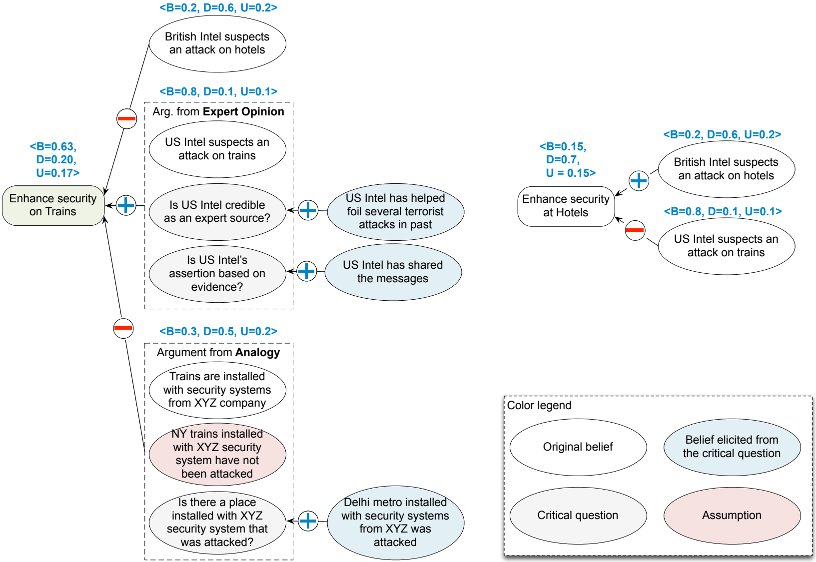
Argument from expert opinion

Major premise	E is an expert in a domain S containing a proposition A
Minor premise	E asserts that proposition A is true (false)
Conclusion	A is true (false)

Critical questions

- What did E assert that implies A?
- Is E's assertion based on evidence?
- How credible is E as an expert?
- Is A consistent with what other experts assert?

Case Study: Arg-ACH Solution (Partial)



Arg-ACH: Empirical Evaluation

What benefits does Arg-ACH offer over ACH?

An end-user study

Subjects 20 (15 graduate and five undergraduate students)

Task Resolve conflicts in the enhancing security scenario. Elicit the necessary beliefs by searching a belief database using keywords

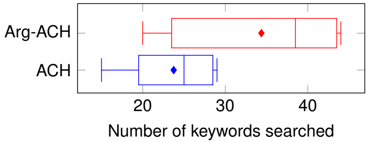
Tools ACH subjects used the PARC ACH tool; Arg-ACH subjects used the Carneades editor [Gordon, 2014] for constructing the argument graphs

Arg-ACH User Study: Results

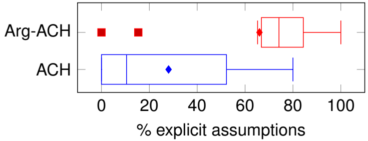
Arg-ACH analysts, compared to ACH analysts

- seek more **evidence** to resolve conflicts
- make more explicit **assumptions**, leaving an audit trail

$\mu_{ACH} < \mu_{Arg-ACH}$ $(p = 0.04) * *$

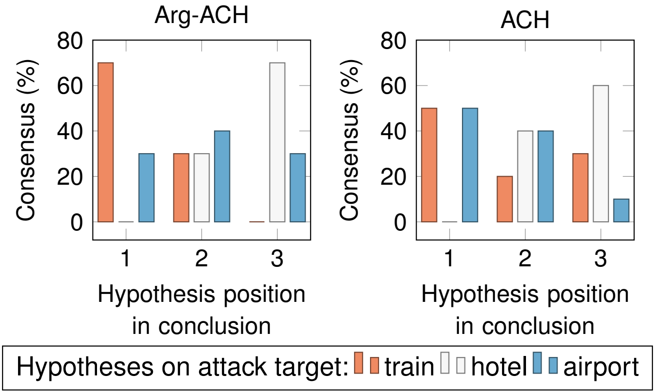


$\mu_{ACH} < \mu_{Arg-ACH}$ $(p = 0.02) * *$



Arg-ACH User Study: Results

- Arg-ACH leads more **consensus** among analysts' conclusion about a conflict compared to ACH



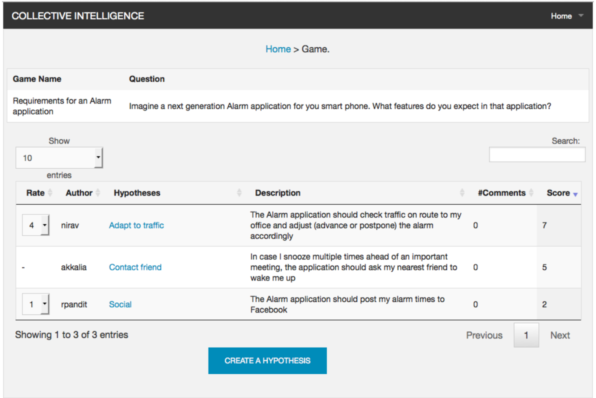
Outline

Facilitating Crowd-RE

How can we elicit creative requirements from the crowd?

- **Data:** To be collected from MTurk
- **Methods:** Well-known, e.g., Delphi, genetic algorithm, personality and creativity surveys
- **Tool:** Developed

Collective Intelligence tool

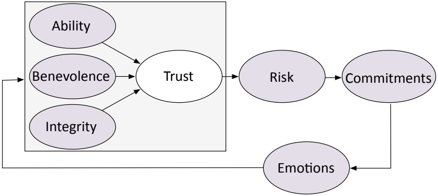


Estimating Trust between Developers

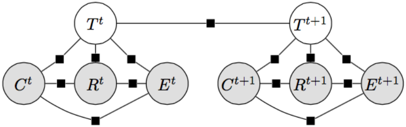
How can we estimate trust between developers in an OSS project?

- **Data:** Available online, e.g., commit logs, and email interactions, and questions-and-answers
- **Method:** Developed [Kalia et al., 2015]

Trust antecedent framework



A CRF-based model



Acknowledgments

Adviser

Dr. Munindar P. Singh, Professor in Computer Science, NCSU

Sponsors

- NSF (Grant 0910868)
- NCSU Laboratory of Analytic Sciences (LAS)

Collaborators

- Anup Kalia, Dr. Chung-Wei Hang, Nirav Ajmeri, Dr. Nithyanthan Poosamani, Dr. Pankaj Telang, and Guangchao Yuan, NCSU
- Dr. Michael N. Huhns, University of South Carolina
- Drs. Tim Finin and Anupam Joshi, UMBC
- Dr. Laura Zavala, CUNY Graduate Center
- Ricard L. Fogues, U. Politecnica de Valencia, Spain
- Dr. Jose Such, Lancaster University, UK

Thank You