

Vision. I envision a computing infrastructure in which autonomous principals interact to realize social applications similar to how human societies operate. Realizing such an infrastructure requires shifting the loci of computation from conceptually centralized servers of today to billions of user-controlled devices (on the Internet of Things). Enabling a truly decentralized infrastructure is a long-term endeavor. In this pursuit, I intend to make fundamental contributions to mobile and social computing, software engineering, artificial intelligence, and human-computer interaction.

Objectives. Mobile social computing is the overarching theme of my research. I recognize *mobile* and *social* factors to be important ingredients of both emerging software *applications* and software engineering *processes* since they both operate in evolving contexts (mobile) and involve multiple stakeholders (social). The key objectives of my research are: (1) to enable mobile social applications to deliver a personalized, privacy-preserving, and context-specific experience to end-users, and (2) to make construction of such applications efficient and effective for developers. To achieve these objectives, I apply techniques from conceptual modeling [4], machine learning [1, 5, 8], argumentation [2], content and social network analysis [3, 7], and crowdsourcing [6].

Empirical methods. I emphasize *empirical* research and have conducted several studies involving human subjects (five published [2, 3, 4, 5] and four ongoing) to evaluate the methods I have been developing. Specifically, I have evaluated (1) software engineering methodologies (e.g., for conceptual modeling, requirements elicitation, and implementation) for their efficiency and repeatability, and (2) artifacts produced by those methodologies (e.g., models, requirements, and software) for their usability (end-user perspective) and comprehensibility (developer perspective).

Big data methods. I have collected large-scale data from sources such Twitter, Foursquare, and Amazon MTurk (e.g., millions of tweets, thousands of MTurk HITs) to evaluate the techniques we have been developing for reasoning about mobile and social factors such as context, sentiment, and trust. We have analyzed these datasets via multiple machine learning paradigms (e.g., active and semi-supervised learning), dynamic models (e.g., conditional random fields), social network analysis (e.g., community detection), and content analysis (e.g., sentiment and topic modeling).

With both empirical and big data skills, I have tried to address several research questions in mobile social computing. Below, I summarize my previous and ongoing works, and plans for near future.

Exploiting Personal Data, Preserving Privacy

Middleware. Emerging software applications are data driven, i.e., they make decisions based on data end-users produce. Although mobile and wearable devices are increasingly capable of sensing a variety of information about end-users' physiology, behavior, and environment, a user may not want to disclose such information to third-parties due to privacy concerns. Further, statistical analysis of data aggregated from multiple users (as it is often done today) may not provide insights objectively valuable to all users. To address these challenges, I developed Platys middleware [5], which, (1) efficiently gathers data from multiple sensors into a user's personal data store shared across multiple user-controlled devices, (2) computes concepts such as places, activities, and social circles specific to the user, and (3) exposes the learned concepts to applications, respecting the user's privacy policies.

Developer study. Platys middleware insulates application developers from dealing with low-level sensors and user-specific idiosyncrasies, and provides an end-user a single point of access to all of his personal data. My findings from an empirical study involving 46 developers indicate that developers employing Platys, compared to Android API, (1) develop applications faster and perceive reduced difficulty, and (2) produce applications that are easier to understand for developers, and potentially more usable and privacy-preserving for application users.

From Conceptual Models to Machine-Learned Models

Agent-oriented methodology. Engineering mobile social application to deliver a user- and context-specific experience is challenging. First, a mobile social application must capture its users' mental models of context—a high-level concern centered on meaning. Second, it must acquire the desired contextual information—a low-level concern centered on devices and infrastructure. I developed Xipho [4], an agent-oriented methodology, for engineering mobile social applications as context-aware personal agents (CPA). Xipho treats context as a cognitive notion and understands other cognitive notions, such as a user's goals and plans, as inherently related to context. Xipho spans all development phases (end-to-end) and leverages reusable components (e.g., a middleware) in a CPA's design and implementation.

Developer study. I evaluated Xipho via a study in which 46 developers employed Xipho or Tropos (a traditional methodology) to engineer three mobile social applications. My findings suggest that Xipho, compared to Tropos, reduces the development time and effort, and yields CPA designs that are easy for developers to comprehend.

Active and semi-supervised machine learning. Deriving useful and user-specific context models from a user’s multimodal (originating from multiple sensors), sparse, and evolving sensor data is challenging. I developed Platys Reasoner [5, 8], an *active* and *semi-supervised* machine learning approach, to map low-level sensor data to high-level contexts such as the user’s places, activities, and social circles. Platys Reasoner offers distinct benefits over the traditional *supervised* (requiring user guidance) and *unsupervised* (not requiring user guidance) learning paradigms.

User study. I employed Platys Reasoner to learn user contexts from user-provided labels and Android phone sensor readings collected from 10 users, finding that Platys Reasoner (1) reduces user effort required for training compared to two traditional supervised approaches, and (2) learns contexts with higher accuracy than two unsupervised approaches since it employs both unlabeled and labeled sensor data.

Mobile Social Applications

I described how my works address the challenges of engineering mobile social applications. Next, I describe some innovative ways in which mobile and social factors can benefit software engineering processes.

Context. I have worked extensively on reasoning about a user’s mobile (e.g., places and activities) and social (e.g., circles and tie strengths) contexts, and how they influence each other, e.g., (1) a user’s places can predict his social circles [3], and (2) conversely, his communities can predict his locations (paper under review).

Applications. Knowing the contexts in which end-users employ applications can assist developers to interpret bugs, reviews, and feature requests in context, e.g., is the negative reviews for the application because it is difficult to use the application from trains? Further, knowing a developer’s contexts can also be valuable, e.g., to recommend context-specific tools, to suggest other developers to connect with, e.g., considering their activities.

Trust. Trust is a crucial factor is achieving high-quality collaboration. We developed TRACE [1], a computational model that predicts trust between users by observing the users’ social interactions and emotions. TRACE employs conditional random fields, and our evaluation on an dataset (collected by IARPA) suggests that TRACE yields lower trust prediction error than traditional hidden Markov models of trust.

Applications. TRACE can be employed to infer trust between users in organizations, social networks, and open source communities by observing users interactions e.g., on discussion forums and email threads. Predicting trust can help build teams, predict performance, and strengthen relationships between developers.

Conflicts. Conflicts are inevitable when multiple stakeholders are involved. We developed the Arg-ACH methodology [2] to resolve conflicts based on stakeholders’ preferences, and more importantly, their arguments for and against those preferences. Arg-ACH builds on the *analysis of competing hypotheses* (ACH), a well-known analytic technique. On a similar line, I am working on inferring privacy policies for content on social media based on stakeholders’ arguments.

Applications. We applied Arg-ACH to resolve goal conflicts that arise during requirements elicitation [2]. The need for resolving conflicts is even more prominent when software engineering processes seek mass participation. Thus, techniques such as Arg-ACH can be of immense value in crowdsourcing software development.

References

- [1] AK Kalia, PK Murukannaiah, and MP Singh. TRACE: A dynamic model of trust for people-driven service engagements. In *Proc. ICSOC*, pp. 353–361, 2015.
- [2] PK Murukannaiah, AK Kalia, PR Telang, and MP Singh. Resolving goal conflicts via argumentation-based analysis of competing hypotheses. In *Proc. RE*, pp. 156–165, 2015.
- [3] PK Murukannaiah and MP Singh. Platys Social: Relating shared places and private social circles. *IEEE Internet Computing*, 16(3):53–59, 2012.
- [4] PK Murukannaiah and MP Singh. Xipho: Extending Tropos to engineer context-aware personal agents. In *Proc. AAMAS*, pp. 309–316, 2014.
- [5] PK Murukannaiah and MP Singh. Platys: An active learning framework for place-aware application development and its evaluation. *ACM TOSEM*, 24(3):1–33, 2015.
- [6] PK Murukannaiah, N Ajmeri, and MP Singh. Acquiring creative requirements from the crowd: Understanding the influences of individual personality and creative potential in Crowd-RE. To appear in *Proc. RE*, pp. 1–10, 2016.
- [7] G Yuan, PK Murukannaiah, Z Zhang, and MP Singh. Exploiting sentiment homophily for link prediction. In *Proc. RecSys*, pages 17–24, 2014.
- [8] L Zavala, PK Murukannaiah, N Poosamani, T Finin, A Joshi, I Rhee, and MP Singh. Platys: From position to place-oriented mobile computing. *AI Magazine*, 36(2):50–62, 2015.