Reasoning about Context and Engineering Context-Aware Agents

(Doctoral Consortium)

Pradeep K. Murukannaiah
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
pmuruka@ncsu.edu

1. INTRODUCTION

A context-aware agent adapts to its human user's *context*—a snapshot of the user's environment, actions, and interactions. Engineering such agents is nontrivial. A context-aware agent must (1) capture its user's mental model of context—a high-level concern centered on meaning, and (2) learn to recognize contexts from sensor data—a low-level concern centered on devices and infrastructure. This thesis develops:

Xipho: an agent-oriented methodology that assists a developer in systematically modeling a context-aware agent via cognitive constructs, and

Platys: a middleware that (1) elicits a subjective context model for each user, (2) learns the context model from sensor data, and (3) shares the user's context across multiple agents in a privacy-preserving manner.

2. XIPHO

We treat context as a cognitive notion and understand other cognitive notions, such as goals and plans, as inherently related to context. A natural approach to developing a context-aware agent would be a methodology that employs cognitive notions throughout development. Existing agent-oriented software engineering methodologies, e.g., [1, 7], describe generic steps of software development, but fall short in dealing with challenges specific to context-aware agents. Xipho [5] fills this gap by providing systematic steps for (1) capturing contextual requirements, (2) deriving an application-specific context information model, and (3) leveraging reusable components in agent implementation.

Xipho guides a developer to explicitly incorporate a user's contexts as beliefs and resources in an agent's high-level model. Specifically, a developer performs three major tasks.

Context-means analysis to identify relationships between contextual beliefs and resources, and goals and plans.

Context information modeling to tailor a generic context metamodel (e.g., place metamodel [6]) to the specific scenarios identified by context-means analysis.

Contextual capability modeling to specify an agent's capabilities to elicit a user's context model, reason about contexts, and trigger context-dependent plans.

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

Figure 1 shows a Xipho model of *Ringer Manager*, a context-aware agent that automatically manages the ringer mode on a user's cell phone based on the user's context. A Xipho model can be specified as a set of contextual capabilities, e.g., $Activity = busy \wedge Proximity = Near \rightarrow Set\ as\ silent$.

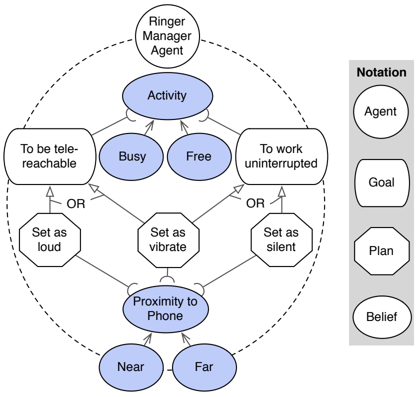

Figure 1: A Xipho model of a context-aware agent.

2.1 Empirical Developer Study

We claim that Xipho (1) reduces *time* and *effort* required to model a context-aware agent, and (2) enhances the *comprehensibility* context-aware agent models thus produced. We empirically evaluated Xipho (X) against the Tropos (T) baseline via a study in which 46 developers modeled three context-aware agents. Figure 2 summarizes our findings.

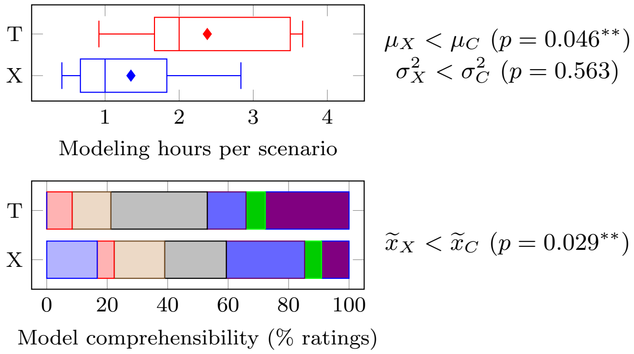

Figure 2: Modeling time and comprehensibility.

3. PLATYS

Xipho yields a high-level context information model specific to an application. However, a context-aware agent must reason about such a model (e.g., if the user is *busy*) from low-level sensor data. Platys middleware separates the concern of context recognition from application development.

Existing techniques, e.g., [2, 8], (1) recognize context objectively (e.g., *home*, *office*, and *elsewhere*), (2) require optimal values for place parameters such as *radius* and *length of stay*, or (3) require frequent sensor readings. As shown in Figure 3, Platys seeks to address these limitations via two complementary machine learning paradigms [3, 6].

Active learning to learn subjective context models, yet requiring as little user effort (for labeling) as possible.

Semi-supervised learning to exploit sporadic data from multiple sensors, given a few labeled readings.

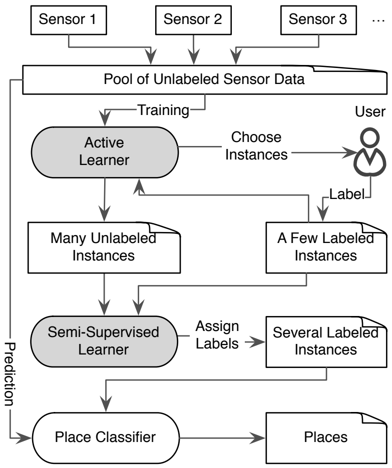

Figure 3: Platys' context-recognition pipeline.

3.1 User Study: Learning from Real Traces

We evaluated Platys via real traces collected from 10 users. Each user carried an Android phone installed with Platys middleware as his primary phone for three to 10 weeks. The middleware collected a user's context labels and recorded GPS, accelerometer, WiFi, and Bluetooth sensor readings.

Figure 4 summarizes our findings that Platys performs (1) on par with a baseline classifier even with significantly fewer labels, and (2) better than a staypoint approach with optimally tuned place parameters.

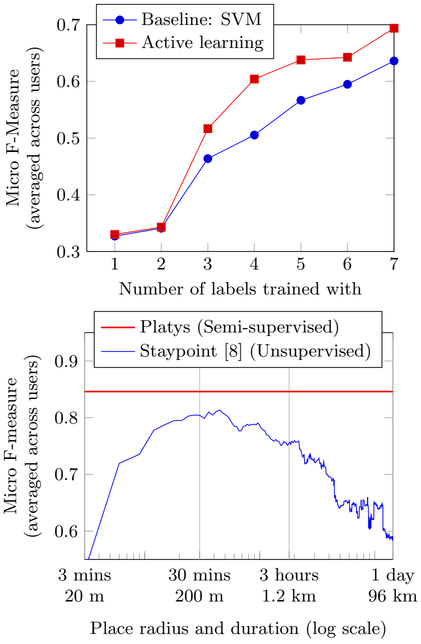

Figure 4: Platys compared with a baseline classifier and an unsupervised staypoint approach.

4. DIRECTIONS

A natural topic for future research is to extend Xipho and Platys to multiagent settings in which context-aware agents can interact to (1) jointly compute context, and (2) make decisions based on each other's contexts, e.g., Ringer Manager can check that the user's neighbors are not *busy* before ringing *loud*. In this direction, we developed Platys Social [4] to identify a user's ego-centric social circles based on shared contexts. Next, we seek to address the following challenges.

Privacy and trust: How can an agent share context information (labels or sensor readings) with trustworthy agents, respecting users' privacy preferences?

Incentive mechanisms: How can we incentivize users to jointly compute context? The joint computation may reduce user effort (e.g., for labeling) and computational resources (e.g., battery consumed).

Sharing tacit knowledge: What is an appropriate mechanism for sharing tacit context knowledge? An ontology may not capture subjective nuances of context and a folksonomy provides little support for reasoning.

5. REFERENCES

[1] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *JAAMAS*, 8(3):203–236, May 2004.

[2] N. Eagle and A. (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, Mar. 2006.

[3] C.-W. Hang, P. K. Murukannaiah, and M. P. Singh. Platys: User-centric place recognition. In *AAAI Workshop on Activity Context-Aware Systems*, 2013.

[4] P. K. Murukannaiah and M. P. Singh. Platys Social: Relating shared places and private social circles. *IEEE Internet Computing*, 16(3):53–59, May 2012.

[5] P. K. Murukannaiah and M. P. Singh. Xipho: Extending Tropos to engineer context-aware personal agents. In *Proc. AAMAS*, 2014.

[6] P. K. Murukannaiah and M. P. Singh. Platys: An empirically evaluated middleware for place-aware application development. TR 2014-2, North Carolina State University, Feb. 2014.

[7] M. Winikoff and L. Padgham. *Developing Intelligent Agent Systems: A Practical Guide*. Halsted Press, New York, 2004.

[8] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending friends and locations based on individual location history. *ACM TWEB*, 5(1):5:1–5:29, Feb. 2011.