

APE: An Annotation Language and Middleware for Energy-Efficient Mobile App Development Paper for CS5371 Soft Test for Mobile & Emb Sys

Jeremy Solmonson
School of Security Engineering
University of Colorado at Colorado Springs
Colorado Springs, CO 80922
Email: jsolmons@uccs.edu

Abstract—This is paper is a critique of "APE: An Annotation Language and Middleware for Energy-Efficient Mobile Application Development." In accordance with homework 2 requirements the critique will be based on: suggestion for acceptance, summary of paper, evaluation of paper, positive points, negatives points, and potential future work.

I. SUGGESTION FOR ACCEPTANCE

I would strongly recommend this paper. The contributions are directly applicable to challenges within current mobile application development. While the experiment could be expanded to additional applications to provide a stronger sample size, this limitation should not hinder publication.

II. SUMMARY OF PAPER

Continuous Runtime Mobile (CRM) applications consume a larger volume of battery life due to their persistent need for resources. The current methods to provide efficient battery consumption are implemented by the programmer in low-level (controlling the individual resource) and system level (synchronizing between resources) optimizations. By deferring resource use to more efficient times (defined by the programmer), batter life can achieve higher longevity compared to no coordination of resources. APE employs Java Annotations to provides in-code modifiers to adjust energy consumption. This allows the programmer to provide adjustments to resources based on events. Employing this technique helped lower energy consumption in an application Citisense.

III. EVALUATION

This paper provides a valuable contribution to the energy savings research. Allowing the programmer to access a standardized API to control power consumption is both usable and efficient.

While this paper does provide a valuable contribution to the mobile energy efficient research, the evaluation only revolves around a single application, Citisense. As Citisense was also developed by the authors, it begs the question if APE was only developed to support the Citisense application. If so, this explains the impressive results of energy efficiency. While I do not think that was the authors intent, and this doesn't seem

to impede the papers contributions/results, it would be more impartial to test on a separate application. As a result, this may not be considered "empirical evidence."

Further the test cases were extremely limited. The application was tested with immediate wifi connectivity - with and without APE. The tests did not include situations with no wifi nor mobile data service was available. This would be beneficial to see the energy efficiencies in non connected environments. Further partial connected environments could be examined with quantifiable weak mobile data service and strong mobile data service. Also, APE enhanced applications could be given to participants to use the mobile devices under real world conditions. These experiments would provide additional validity to the authors contributions.

The impressive portion of this paper is the ability to logically design energy efficient algorithms using finite state machines. Once the most efficient algorithm is found, then the programmer can develop the software to meet the design. The designs can be compared an contrasted to find the most efficient conditions per mobile device.

IV. POSITIVE POINTS

- + Provides real world benefits that programmers could implement immediately to increase battery efficiency.
- + The ability to design a finite state machine and provide functional coding examples is a plus. Good use of examples.
- + Recent research is well cited and compares/contrasts to the proposed APE enhancement.

V. NEGATIVE POINTS

- Small sample size (Citisense) to test the APE framework. Unknown data on the actual real-world battery efficiency.
- More test cases should be examined.

VI. POTENTIAL FUTURE WORK

The first area of research would be to validate the APE functions in real world conditions. If APE drains the batter in austerer conditions (such as mountain ranges with limited connectivity) the this solution isn't practical. The need to

verify the author's contributions for additional environments is needed.

The other main area for future research is expanding the APE framework for optimization. This will help answer questions similar to, "what is the optimal time a resource should wait before sending data (1 ms, 1 sec, 1 min, 1 hour, etc.)?" The API can be expanded to support additional events or resource synchronization. Further, research can be explored to provide the most energy efficient conditions. These conditions can be compared against each other with standardized results.