UNIVERSITÄT BERN

Investigating Energy-Related Practices in Web Applications

Veronika Kousadianos

Supervised by Dr. Pooja Rani
and Prof. Dr. Timo Kehrer

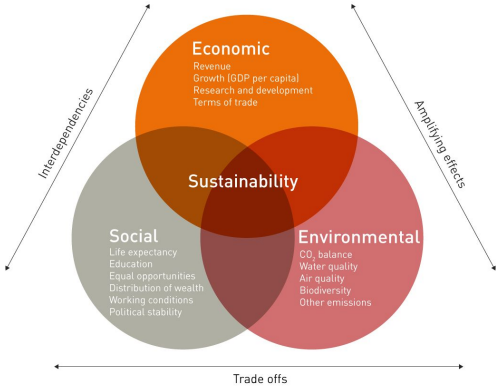
Software Engineering Group
University of Bern, Switzerland

Content

Motivation

Def: “Sustainability means meeting our own needs without compromising the ability of future generations to meet their own needs.” (McGill University, 2013)

The three dimensions of sustainability and a selection of indicators



Source: own presentation, based on Passet (1979)
www.economiesuisse.ch

https://www.economiesuisse.ch/en/dossier-politics/sustainable-development-includes-several-dimensions, 28.5.22

Sustainability in Software Engineering?

Software Sustainability

Goal: to achieve sustainability in software

(Calero et al, 2022)

Sustainability in Software Engineering

Software Sustainability

Goal: to achieve sustainability in software

- Social Sustainability
- Economic Sustainability
- Environmental Sustainability

Software as Part of Sustainability

Considers software as a new dimension of sustainability.

- Individual Sustainability
- Social Sustainability
- Economic Sustainability
- Environmental Sustainability
- Technical Sustainability

(Calero et al, 2022)

Motivation

Def. Green and Sustainable Software Process:

“Software that meets its (realistic) sustainability objectives, expressed in terms of direct and indirect impacts on economy, society, human beings, and environment that result from its definition and deployment.” (Mahmoud et al, 2013)

https://de.wikipedia.org/wiki/Energy_Star (07.03.2023)

Methodology

Catalog of Energy Patterns for Mobile Application by Cruz and Abreu (2019)

- Main Contribution: 22 energy patterns for mobile apps

Methodology

- App dataset collection
- Thematic analysis

Methodology

Catalog of Energy Patterns for Mobile Application by Cruz and Abreu (2019)

- Main Contribution: 22 energy patterns for mobile apps

Methodology

1. App dataset collection
   a. 1779 open source projects
2. Automatic gathering of subjects with potential interest
   a. .*(energy|battery|power).*
   b. Collected 6028 subjects
3. Manual refinement of subjects of interest
   a. Reduced number of subjects to 1563
4. Thematic analysis
   a. Familiarization with data
   b. Generating initial labels
   c. Reviewing themes
   d. Defining and naming themes

(Cruz et al, 2019)

Catalog of Energy Patterns for Mobile Application

Dark UI Colors

“Provide a dark UI color theme to save battery on devices with AMOLED screens.”

Power Awareness

“Have a different behavior when the device is connected/disconnected to a power station or has different battery levels.”

Race-to-idle

“Release resources or services as soon as possible (e.g., wakelocks, screen).”

(Cruz et al, 2019)

Catalog of Energy Patterns for Mobile Applications

1. Dark UI Colors:
   Provide a dark UI color theme to save battery on devices with AMOLED screens.
2. Dynamic Retry Delay:
   Whenever an attempt to access a resource has failed, increase the interval of time waited before asking access to that same resource.
3. Avoid Extraneous Work:
   Avoid performing tasks that are not visible/valuable to the user and/or quickly become obsolete.
4. Race-to-idle:
   Release resources or services as soon as possible (e.g., wakelocks, screen)
5. Open Only When Necessary:
   Open/start resources/services only when they are strictly necessary.
6. Push Over Poll:
   Use push notifications to receive updates from resources, instead of actively querying resources (i.e., polling).
7. Power Save Mode:
   Provide an energy efficient mode in which user experience can drop for the sake of better energy usage.
8. Power Awareness:
   Have a different behavior when a device is connected/disconnected to a power station, or has different battery levels.
9. Reduce Size:
   When transmitting data, reduce its size as much as possible.
10. WiFi Over Cellular:
    Delay or disable heavy data connections until the device is connected to a WiFi network.
11. Suppress Logs:
    Avoid using intensive logging (< 1Hz).
12. Batch Operations:
    Batch multiple operations instead of putting the device into an active state many times.
13. Cache:
    Avoid performing unnecessary operations by using cache mechanisms.
14. Decrease Rate:
    Increase time between syncs/sensor reads as much as possible.
15. User Knows Best:
    Allow users to enable/disable certain features in order to save energy.
16. Inform Users:
    Let the user know if the app is doing any battery intensive operation.
17. Enough resolution:
    Collect or provide high accuracy data only when strictly necessary.
18. Sensor Fusion:
    Use data from low power sensors to infer whether new data needs to be collected from high power sensors
19. Kill Abnormal Tasks:
    Provide means of interrupting energy greedy operations (e.g., using timeouts, or users input)
20. No screen interaction:
    Whenever possible allow interaction without using the display.
21. Avoid Extraneous Graphics and Animations:
    Graphics and animations are really important to improve user experience. However, they can also be battery intensive – use them with moderation
22. Manual Sync - On Demand:
    Perform tasks only when the user specifically asks.

(Cruz et al, 2019)

Dark UI Colors

“Provide a dark UI color theme to save battery on devices with AMOLED screens.” (Cruz et al, 2019)

Adaptation to web?

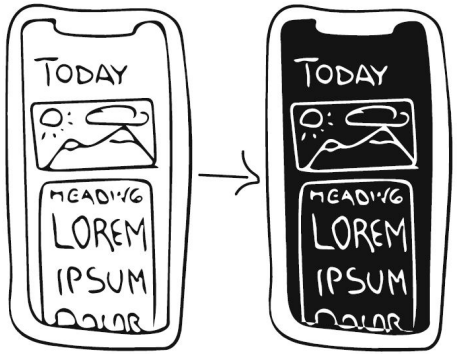


Fig. 4 UI themes with dark colors are more energy efficient

Dark UI Colors

“Provide a dark UI color theme to save battery on devices with AMOLED screens.” (Cruz et al, 2019)

Adaptation to web

Solution: Provide a UI with dark background colors.

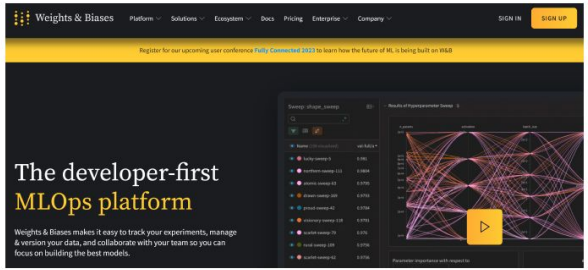


Figure 3.1: Screenshot of a website with dark UI colors.

Race-to-idle

“Release resources or services as soon as possible (e.g., wakelocks, screen).” (Cruz et al, 2019)

Adaptation to web?

Race-to-idle

“Release resources or services as soon as possible (e.g., wakelocks, screen).” (Cruz et al, 2019)

Adaptation to web

Solution: Release resources and services as soon as they are not needed anymore.

Next Steps

- Complete pattern transfer from mobile to web

- Use codebase of Smart Energy Link as case study
  - If unsuccessful, look for open source projects

Smart Energy Link

Smart Energy Link focuses on measuring solar power in ZEV (Zusammenschluss Eigenverbrauch, engl. own consumption community) and optimizing energy consumption.

https://smartenergylink.ch/de/

SEL SMART ENERGY LINK

Code Example

Race-to-idle

“Release resources or services as soon as possible.” (Cruz et al, 2019)

```
try:
    cursor = connection.cursor()
    cursor.execute(
        '''SELECT id, community_id, valid_from, valid_to, provider_costs, energy_delivery_costs,
         solar_feedin_yield, measuring_costs, administrative_costs, initial_costs, operational_costs,
         grid_consumption, high_consumption, low_consumption, solar_consumption, total_solar_production
         FROM backend_powercosts WHERE
        community_id = {} AND valid_to IS NULL ORDER BY valid_to DESC
        '''.format(community.id))
    power_costs_valid_to_null = cursor.fetchall()
```

cursor.close() # not used

Challenges

- Lack of tools to measure energy consumption
- Lack of awareness in development
- Lack of research in web applications

- very broad topic

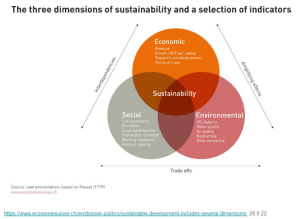
Future Work

- Automate the process of finding (anti-) patterns in source code
- What is the impact of applying energy patterns in web application?
- How do different websites vary in terms of energy consumption?

Summary

Motivation

Def: “Sustainability means meeting our own needs without compromising the ability of future generations to meet their own needs.” (McGill University, 2013)



Sustainability in Software Engineering?

Software Sustainability
Goal: to achieve sustainability in software

Software as Part of Sustainability
Considers software as a new dimension of sustainability.

(Calero et al, 2022)

Catalog of Energy Patterns for Mobile Application

Dark UI Colors

“Provide a dark UI color theme to save battery on devices with AMOLED screens.”

Power Awareness

“Have a different behavior when the device is connected/disconnected to a power station or has different battery levels.”

Race-to-idle

“Release resources or services as soon as possible (e.g., wakelocks, screen).”

(Cruz et al, 2019)

Race-to-idle

“Release resources or services as soon as possible (e.g., wakelocks, screen).” (Cruz et al, 2019)

Adaptation to web

Solution: Release resources and services as soon as they are not needed anymore.

Code Example

Race-to-idle
“Release resources or services as soon as possible.” (Cruz et al, 2019)

```
try:
    cursor = connection.cursor()
    cursor.execute(
        '''SELECT id, community_id, valid_from, valid_to, provider_costs, energy_delivery_costs,
        solar_feedin_yield, measuring_costs, administrative_costs, initial_costs, operational_costs,
        grid_consumption, high_consumption, low_consumption, solar_consumption, total_solar_production
        FROM backend_powercosts WHERE
        community_id = {} AND valid_to IS NULL ORDER BY valid_to DESC
        '''.format(community.id))
    power_costs_valid_to_null = cursor.fetchall()
```

cursor.close() # not used

Sources

C. Calero, M. Ángeles Moraga and F. García, "Software, Sustainability, and UN Sustainable Development Goals," in IT Professional, vol. 24, no. 1, pp. 41-48, 1 Jan.-Feb. 2022, doi: 10.1109/MITP.2021.3117344.

L. Cruz and R. Abreu. Catalog of energy patterns for mobile applications. Empirical Software Engineering, 24(4):2209–2235, 2019.

S.S. Mahmoud, I. Ahmad, "A Green Model for Sustainable Software Engineering", International Journal of Software Engineering and its Applications, January 2013, 7(4):55-74.