

WarpMind #4 Assignment Report: The Paths of Refinement Option 1: CSS and JavaScript frameworks

Adrián Fernández-Paniagua Belinchón

Grazia Cossu

DA4 5

October 3, 2025

Contents

1	Introduction	2
2	Refactoring Process	2
2.1	Choosing and Integrating the Frameworks	2
2.2	Rebuilding the Layout	2
2.3	Refactoring JavaScript with jQuery	2
3	Reflections	3
3.1	Positive Aspects	3
3.2	Challenges and Less Positive Experiences	3
4	User Interface Before and After the Refactoring	3
4.1	Original Interface	3
4.2	New Layout with Bootstrap	4
4.3	Concept Extraction and Card Rendering	4
4.4	Responsive Behaviour	5
5	Conclusion	8

1 Introduction

For this assignment I decided to rework my WarpMind interface by introducing one CSS framework and one JavaScript client-side framework. I chose **Bootstrap** and **jQuery** because they are well-established, easy to integrate into a small project, and flexible enough to reshape the interface without forcing a full rewrite. My goal was not only to modernise the appearance of the tool, but also to make the code easier to maintain and to improve the experience across different devices.

2 Refactoring Process

2.1 Choosing and Integrating the Frameworks

The first step was simply getting the frameworks into the project. Once I added the Bootstrap and jQuery CDNs to the `index.html`, the entire interface immediately changed character. Bootstrap brought a much more uniform style than my handwritten CSS, and I quickly realised that many of my custom rules were no longer necessary. jQuery, instead, felt like adding a layer of convenience over the functions I had already implemented. Even small changes, like switching to `$(#id)` selectors, made the code feel lighter and more coherent.

2.2 Rebuilding the Layout

What took the most time was reorganising the layout to use Bootstrap's grid system. The original interface worked, but it was rigid and definitely not friendly on smaller screens. With Bootstrap I structured the page into a simple three-column layout: the concept list on the left, the PDF viewer in the centre, and the workspace on the right.

The surprising part was how easy it became to make everything responsive. Once I replaced my fixed widths with Bootstrap's `col-` classes, the interface naturally collapsed into a vertical layout on a phone without further tweaking. Even simple components like buttons and sliders instantly looked cleaner once wrapped in Bootstrap classes.

2.3 Refactoring JavaScript with jQuery

Moving the JavaScript logic to jQuery required more thought. A lot of my code interacted with elements that were created dynamically from a template, and these elements did not respond to event listeners in the same way as static ones. I had to rethink how interactions were triggered and eventually adopted jQuery's event delegation pattern.

Although this initially felt counterintuitive, it ended up simplifying the entire script. Instead of scattering listeners everywhere, I could attach a single listener to a parent element and let jQuery handle the rest. The result was cleaner, more compact, and easier to read.

3 Reflections

3.1 Positive Aspects

What I appreciated most in this refactoring is the immediate sense of order that frameworks bring. Bootstrap removed the visual roughness of the prototype and gave me a layout that behaves well even in edge cases. jQuery, on the other hand, encouraged me to restructure parts of my script in a more thoughtful way, especially regarding dynamic elements.

Another unexpected benefit was how quickly the interface started to feel like a more mature application. Even though the underlying logic did not change dramatically, the combination of better styling and clearer interaction patterns made the tool feel more professional.

3.2 Challenges and Less Positive Experiences

The main difficulty emerged from the combination of old and new code. As soon as Bootstrap's styles became active, several of my previous CSS rules started clashing with it, creating some awkward layouts. It took patience to gradually remove or adapt the conflicting parts without breaking the entire interface.

The switch to jQuery also required learning some habits from scratch. Event delegation in particular took a while to understand, because the behaviour is less explicit than directly attaching event listeners. Debugging dynamic behaviour in the workspace section was the most time-consuming part of the refactor.

4 User Interface Before and After the Refactoring

To better illustrate how Bootstrap and jQuery changed the interface, I compared the previous layout with the refactored version. The screenshots below show both the original state of the tool and the redesigned version.

4.1 Original Interface

The initial interface relied mainly on custom CSS with a fixed layout, which made it less adaptable and visually inconsistent. Figure 1 shows a typical view of the older version.

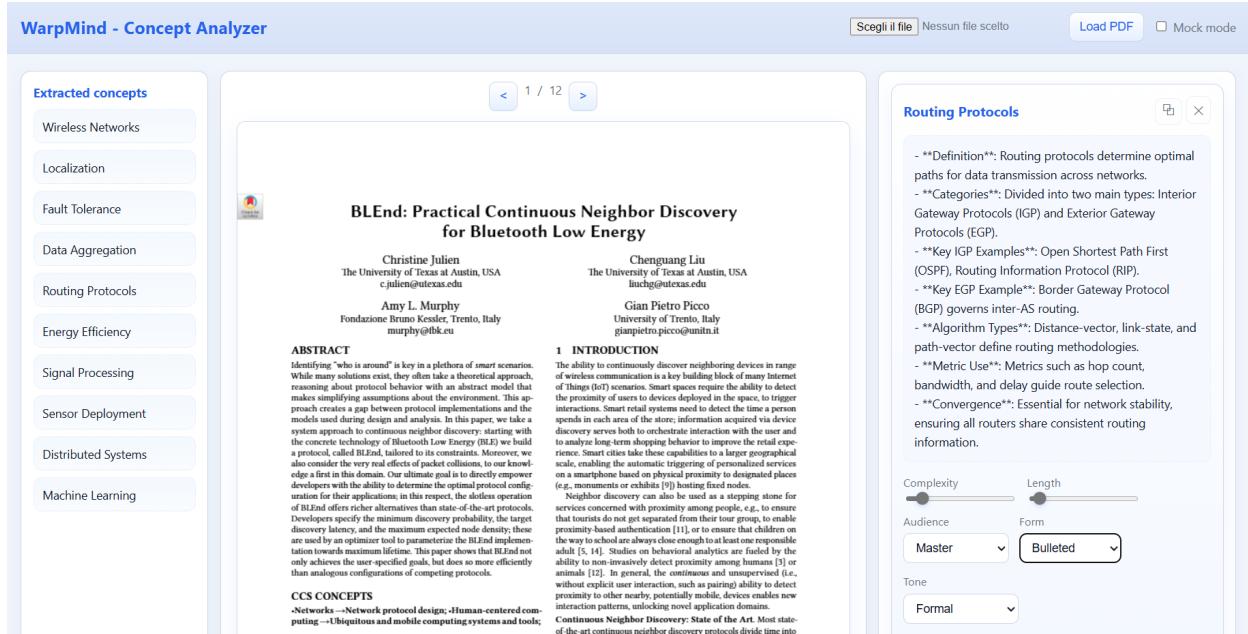


Figure 1: Original WarpMind interface before refactoring, based entirely on handwritten CSS.

4.2 New Layout with Bootstrap

Figure 2 displays the updated interface after integrating Bootstrap. The three-column layout is now handled by the grid system, making the interface more balanced and immediately more readable.

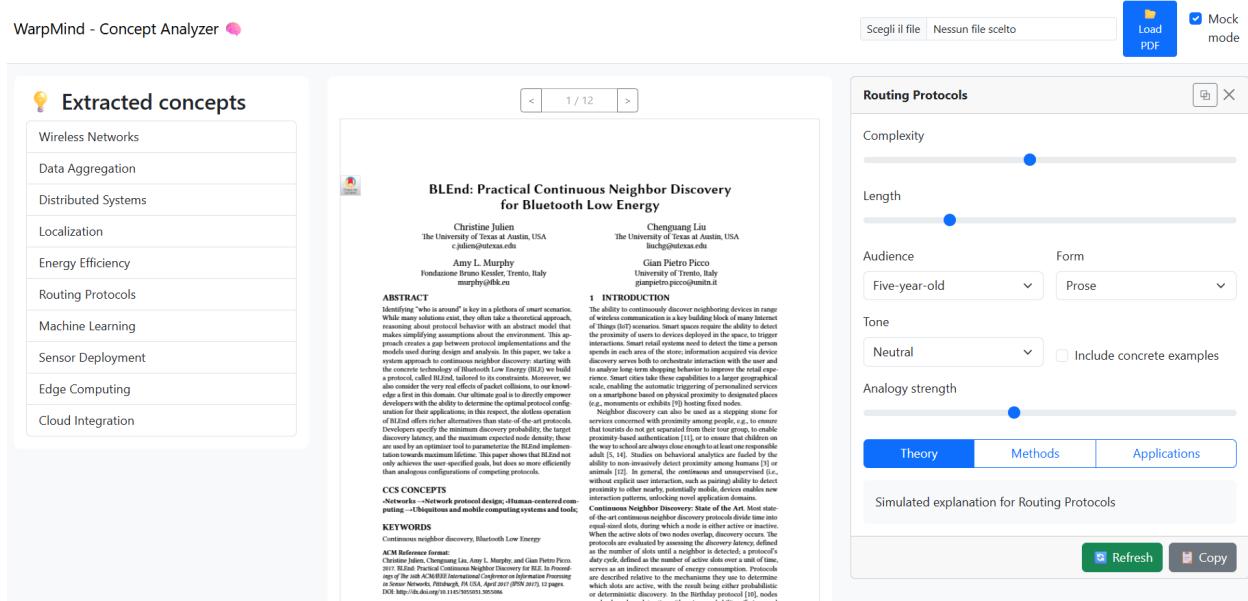


Figure 2: Refactored WarpMind interface using Bootstrap's grid system and improved spacing.

4.3 Concept Extraction and Card Rendering

Bootstrap list-groups and cards allowed for a cleaner and more structured presentation of dynamically generated content. As shown in Figure 3, the extracted concepts appear neatly in the sidebar, and clicking on them produces an interactive card.

The figure consists of two main parts. On the left, there is a sidebar titled "Extracted concepts" containing a list of terms such as Joule-heated reaction field, spiral-shaped catalyst, CO₂ methanation, infrared thermal imaging, thermal management, carbon capture utilization, hydrogenation, swirl flow, methane production, and energy efficiency. Below this list is a note indicating the article is licensed under a Creative Commons Attribution 3.0 Unported Licence. On the right, there is a detailed concept card for a research paper from RSC Advances. The card includes the title "e-Methanation with a spiral catalyst: optimized thermal management and long-term stability†", authors (Ryo Watanabe, Kenjiro Nishida, Hirofumi Suganuma, Priyanka Verma), and the journal information (RSC Advances, 2020, 15, 736). The card also contains the abstract, introduction, and several sections of the paper's content. To the right of the card is a "infrared thermal imaging" interface with various filters and settings.

Figure 3: Extracted concepts (left) and example of a dynamically generated concept card (right).

4.4 Responsive Behaviour

Bootstrap's responsiveness significantly improved usability on smaller screens. Figures 4 and 5 show how the interface reorganises itself without requiring additional CSS.

WarpMind - Concept Analyzer 💬

Scegli il file Nessun file scelto

Mock mode

💡 Extracted concepts

- Edge Computing
- Wireless Networks
- Data Aggregation
- Cloud Integration
- Signal Processing
- Machine Learning
- Routing Protocols
- Distributed Systems
- Fault Tolerance
- Sensor Deployment

< 1 / 12 >

 BLEnd: Practical Continuous Neighbor Discovery for Bluetooth Low Energy

Figure 4: Medium-sized view: components reorganise while maintaining clarity.

tation towards maximum lifetime. This paper shows that BLEnd not only achieves the user-specified goals, but does so more efficiently than analogous configurations of competing protocols.

CCS CONCEPTS

- Networks → Network protocol design;
- Human-centered computing → Ubiquitous and mobile computing systems and tools;

KEYWORDS

Continuous neighbor discovery, Bluetooth Low Energy

ACM Reference format:

Christine Julien, Chengguang Liu, Amy L. Murphy, and Gian Pietro Picco. 2017. BLEnd: Practical Continuous Neighbor Discovery for BLE. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Pittsburgh, PA USA, April 2017 (IPSN 2017), 12 pages. DOI: <http://dx.doi.org/10.1145/3055013.3055086>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
 IPSN 2017, Pittsburgh, PA USA
 © 2017 ACM. 978-1-4503-4890-4/17/04...\$15.00
 DOI: <http://dx.doi.org/10.1145/3055013.3055086>

adult [5, 14]. Studies on behavioral analytics are fueled by the ability to non-invasively detect proximity among humans [3] or animals [12]. In general, the continuous and unsupervised (i.e., without explicit user interaction, such as pairing) ability to detect proximity to other nearby, potentially mobile, devices enables new interaction patterns, unlocking novel application domains.

Continuous Neighbor Discovery: State of the Art Most state-of-the-art continuous neighbor discovery protocols divide time into equal-sized slots, during which a node is either active or inactive. When the active slots of two nodes overlap, discovery occurs. The protocols are evaluated by assessing the *discovery latency*, defined as the number of slots until a neighbor is detected; a protocol's *slot cycle*, defined as the number of active slots over a unit of time, serves as an indirect measure of energy consumption. Protocols are described relative to the mechanisms they use to determine which slots are active, with the result being either probabilistic or deterministic discovery. In the Birthday protocol [10], nodes randomly make a slot active with a given probability, offering good average case performance but not providing guarantees on discovery latency. Instead, Discu [4] and U-Connect [7] space active slots according to prime numbers, relying on the properties of the Chinese Remainder Theorem to guarantee discovery within a tight time bound. Searchlight [1] and BlindDate [17] offer hybrid approaches, placing some active slots for deterministic discovery, then adding more in a pseudo-random manner to improve performance.

105

Wireless Networks

✖

Complexity

Length

Audience

Five-year-old

▼

Form

Prose

▼

Tone

Neutral

▼

Figure 5: Mobile-friendly layout where columns stack vertically and buttons adapt.

5 Conclusion

Overall, integrating Bootstrap and jQuery proved to be a very positive experience and gave me a clearer sense of how frontend frameworks can support, rather than complicate, a project. The WarpMind interface is now more polished, more flexible, and easier to extend.

While there were moments where it felt like I was wrestling with the frameworks rather than working with them, the final outcome confirmed that the refactoring effort was worthwhile. The application now behaves more consistently and is better prepared for future improvements, including the possibility of evolving into a full Electron-based desktop tool.