

TD-Browser – A Beginner-friendly Web-Client for the Web of Things

Osama Hanoun¹

Abstract: In this paper, the author introduce the TD-Browser that enables visual interactions with Web Things by generating user interface elements from Thing Descriptions. The target group are beginners who did not yet understand the concept of the Web of Things. Currently, most available tools focus on scientific purposes with a lack of documentation which makes it hard for newcomers to gain practical experience. For evaluation, I conducted a study using the Concurrent Think-aloud method with one subject to uncover first design flaws. Although TD-Browser cannot be used for teaching beginners the concepts of the Web of Things for now, its clean user interface enables users to work and interact effortlessly with Web Things.

Keywords: Prototyping; Semantic Web; Thing Description; Web of Things

1 Introduction

The Web of Things (WoT) is a subset of the Internet of Things (IoT) which aims to provide universal protocol interoperability by leveraging standard Web protocols such as the HyperText Transfer Protocol (HTTP). The W3C WoT Task Force defines three main components for the WoT [Ka20]: (1) A *Thing* (or a Web Thing) which is an abstraction of a physical or a virtual entity, (2) a *Thing Description* (TD) which is both a machine- and human-readable way to describe the metadata and the interface of a Thing, and (3) a so-called *Consumer* which is an entity that can process WoT Thing Descriptions.

Although the first public Working Draft for the WoT-TD was published in 2017², there is still a scarcity of beginner-friendly tools for the Web of Things (WoT) that can be used for prototyping and testing. In order for users to be able to interact with a Thing, it is required from them to develop a Consumer that can process the TD for the targeted Thing. Developing a Consumer requires the comprehension of the WoT architecture³ and proficient programming skills. Therefore, it is challenging for students and scientists who are newcomers, I will describe them as beginners through this paper, to the WoT architecture, whenever they would like to try interacting with published Things or to test a Thing that they have created. As a solution, I introduce TD-Browser, a Web-client and WoT Consumer

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Lange Gasse 20, 90403, Nuremberg, Germany, osama.hanoun@fau.de

² <https://www.w3.org/TR/2017/WD-wot-thing-description-20170914/>, last accessed on April 25th, 2022.

³ <https://www.w3.org/TR/wot-architecture/>, last accessed on April 25th, 2022.

which can generate a graphical user interface (GUI) on the basis of a TD. This allows users to start interacting with a Thing graphically with reduced effort by overcoming most technical barriers and minimizing the required knowledge of the WoT Scripting API⁴ implementation.

2 Related Work

The WoT architecture defines three interaction affordances [Ka20]: (1) Properties, (2) Actions, and (3) Events. Properties expose the state of a Thing, for example, the model name of a device or the current data of a sensor. Actions are used to invoke a function of a Thing, for example, asking a printer to print a document. Events are asynchronous data sent by a thing to all consumers, for example, a Smart Radiator alerting all Consumers when its heat increases.

There are related projects, as mentioned in the developers page on the W3C Web of Things website⁵, such as Web of Things API Development Environment (W-ADE)⁶, WoT-FXUI⁷, and Browsified node-wot⁸. However, all of the mentioned projects do not support the latest WoT standard.

W-ADE [Sc20] provides many features including TDs parsing, interpreting and editing, communicating over various protocols (e.g., MQTT), adding security credentials and providing timing performance testing. Browsified node-wot allows users to read properties, invoke actions, and subscribe to events. But, it has a poorly structured user interface and does not support all the affordance interaction like observing a property and reading all properties. In contrast to the previously mentioned projects, TD-Browser covers all the interaction affordances and provides a beginner friendly user interface as will be discussed in the evaluation section.

3 Design and Implementation

TD-Browser⁹ addresses users with limited knowledge of the WoT, where having a clear user interface plays a significant role especially when it is the user's first interaction with the WoT. Following Don Norman's principles of design (see Table 1), TD-Browser ensures *visibility* of all the features by using clear descriptive buttons that correspond to the WoT Scripting API. Users receive an instant *feedback* for any action that they perform. For example, when TD-Browser consumes a TD, the user will be informed by either a green success message or a red error message. TD-Browser puts *constraints* on users by preventing accessing

⁴ <https://www.w3.org/TR/wot-scripting-api/>, last accessed on April 25th, 2022.

⁵ <https://www.w3.org/WoT/developers/>, last accessed on April 25th, 2022.

⁶ <https://github.com/tum-esi/wade/>, last accessed on April 25th, 2022.

⁷ <https://github.com/danielpeintner/wot-fxui/>, last accessed on April 25th, 2022.

⁸ <http://plugfest.thingweb.io/webui/>, last accessed on April 25th, 2022.

⁹ <https://github.com/wintechis/TD-Browser/>

more than one interaction affordance at a given time. Having representative icons allows TD-Browser to provide an excellent *mapping* between icons and their linked effects. If a user wants to invoke an action or maybe write a property, then the user will take *consistent* steps in order to achieve the goal. In TD-Browser, users will experience consistent color change of buttons when interacting with them. Figure 1 shows a screenshot of TD-Browser.

Tab. 1: The Don Norman principles of design which used for TD-Browser [No02].

Principle	Meaning
Visibility	Ensuring that users are able to find the features that they are looking for and to know what to do next.
Feedback	Providing users with feedback for what action has been invoked and what has been accomplished.
Constraints	Restricting user interactions with the product at a given moment.
Mapping	Implementing design with a clear relationship between controls and the effect they have.
Consistency	Similar tasks should have similar operations and similar elements.
Affordance	An attribute of an object that should give the user a hint on how to use it.

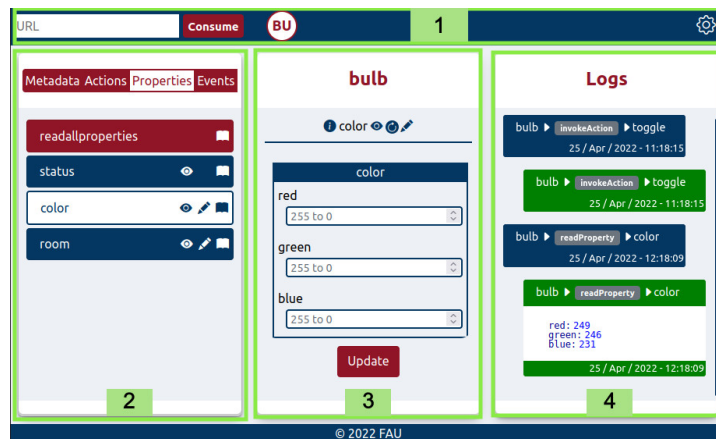


Fig. 1: Screenshot of TD-Browser on desktop showing the main interface components: (1) The Navbar which contains a form for consuming TDs, switching between consumed Things, and changing settings. (2) The Left-frame which contains the metadata, the affordances, and the interaction affordances for the active consumed Thing. (3) The Middle-frame where the user can interact with selected interaction affordance and view the responses from a Thing. (4) The Right-frame where the user can see all requests and responses that happened during a session.

TD-Browser runs on browsers, on both smartphones and desktops, by using web development languages like Hypertext Markup Language revision 5 (HTML5), Cascading Style Sheets level 3 (CSS3), and JavaScript. Also, TD-Browser uses the Eclipse Thingweb node-wot package¹⁰ which is a W3C Web of Things implementation with NodeJS. TD-Browser can

¹⁰ <https://github.com/eclipse/thingweb.node-wot/>, last accessed on April 25th, 2022

be used with no internet connection if a user has the source code and decides to only interact through a local network.

The following scenario will be used to explain the TD-Browser architecture and how it works. A user provides TD-Browser with a TD of a Thing then TD-Browser generates a graphical user interface that corresponds to the provided TD. When the user interacts with an interaction affordance, TD-Browser will convert any entered data and the performed interaction to an acceptable syntax by Eclipse Thingweb node-wot package. After that, the package sends a request to the Thing and when the package receives a response then it will be provided to TD-Browser. In the end, the TD-Browser displays the response to the user.

TD-Browser provides the following main features: (1) Supporting HTTP protocol. (2) Consuming a local or web-based JSON/JSON-LD file. (3) Consuming multiple TDs at the same time. (4) Reading, observing, and writing properties. (5) Invoking actions. (6) Subscribing and unsubscribing to events. (7) Validating inputs according to the consumed TD. (8) Accessible on the browser by both desktop and smartphone with a responsive view.

4 Evaluation

TD-Browser targets beginner users of the WoT with a beginner-friendly interface. In order to test its eligibility for beginners, an evaluation is required. In addition, the evaluation is important to uncover any design flaws and relevant usability elements.

In this evaluation I used the Concurrent Think-aloud method (CTA) [MH12] which is an evaluation method where subjects verbalize their thinking, doing, or feeling while completing a set of predefined tasks. According to an international survey done by McDonald, Edwards, and Zhao [MEZ12] showed that CTA is the most frequently used approach. Also, CTA is easy and fast to conduct.

I recruited one female computer engineering student from Jordan (22 years old), by word of mouth, who did not have prior knowledge of the WoT architecture and was not associated with my research work. Also, she did not receive any compensation and was just participating out of interest. She was told that TD-Browser allows users to control devices connected to the internet by providing the device URI (Uniform Resource Identifier) and she was given a set of tasks as a scenario to complete that covered virtual Thing features. The virtual Thing used in evaluation was a virtual light bulb with interaction affordances which is listed in Table 2. In addition, I provided her with a Web page that simulates the behavior of the virtual light bulb (see Figure 2). The total evaluation lasted one hour.

The subject was given a scenario and a set of tasks as listed in Table 3. The scenario was that the subject gave her grandmother a new smart light bulb as a gift. Then before she goes back home, she connects the light bulb for her grandmother to the Wi-Fi. After she arrived home, her grandmother calls her back in order to help her grandmother. Since, the

Tab. 2: Interaction affordances of the virtual light bulb used in the evaluation

Affordance Class	Interaction Affordance	Description
Action	toggle	Toggle the light bulb (on off)
	reset	Reset the light bulb settings
Property	status	Current status of the light bulb (on off)
	color	Current RGB color of the light bulb
	room	Current Location of the light bulb
Event	power_on	Fire the event when the power is back on

light bulb can be controlled by TD-Browser from her home, therefore, she shall help her grandmother by completing the tasks.

The subject was able to complete all the tasks without any problems. As I observed during the evaluation, the subject would solve a task by looking for mutual words between a task and the TD-Browser interface. In case no mutual words were found, she would hover over the icons in the TD-Browser interface and read their tooltip. The approach that the subject used to complete the tasks relies on the clarity of the TD-browser interface and its self-descriptive buttons and icons. Regarding tasks 2, 9, 13 and 14 in Table 3, even though the subject did not face any problems in completing these tasks, she had trouble to understand the phrased tasks with the WoT terminology. In summary, she could interact with WoT Things without understanding the WoT architecture. Therefore, I conclude that TD-Browser can be valuable to users who recently have started with the WoT, but TD-Browser does not teach users the concept of the WoT architecture. Future work should focus on developing a didactic concept to teach the WoT architecture to beginners while interacting with TD-Browser.

5 Conclusion

In order for more people to adhere to the W3C standards of WoT, more development tools and packages are needed to encourage them to adapt these standards. Users who are new to the Web of Things environment struggle to get started with available tools due to their lacking documentation. To support beginners, I introduced the TD-Browser which enables visual interactions with Web Things. I conducted a study using the Concurrent Think-aloud method with one subject to uncover first design flaws. Although TD-Browser cannot be used for teaching beginners the concepts of the Web of Things, its clean user interface enables users to work and interact with Web Things nonetheless. Future work aims to close the gap between applying the WoT and understanding it.

Tab. 3: The tasks which were used for the evaluation

#	Task
1	Provide the app with the URI that was given to you in order to gain control over the light bulb.
2	State the type of security that is used for the light bulb.
3	State the available actions.
4	Show the description of the toggle action.
5	Invoke the toggle action.
6	State the available properties.
7	Read the status property.
8	Change the color of the light bulb to yellow (R:255, G:255,B:0).
9	Observe the status property.
10	Change the location of the bulb from the room property to “bedroom”.
11	State the available events.
12	Subscribe to the power_on event.
13	Unsubscribe to the power_on event.

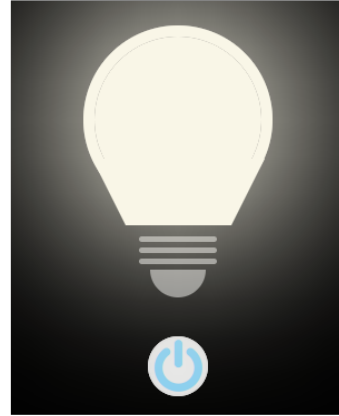


Fig. 2: A screenshot of the virtual light bulb used in the evaluation

6 Acknowledgements

I would like to thank the Chair of Technical Information Systems at Friedrich-Alexander-Universität Erlangen-Nürnberg. Without the support of the Chair this paper and this project would not have been possible.

Bibliography

- [Ka20] Kajimoto, Kazuo; Lagally, Michael; Kawaguchi, Toru; Matsukura, Ryuichi; Toumura, Kunihiro: Web of Things (WoT) Architecture 1.1. W3C working draft, W3C, November 2020. <https://www.w3.org/TR/2020/WD-wot-architecture11-20201124/>.
- [MEZ12] McDonald, Sharon; Edwards, Helen; Zhao, Tingting: Exploring Think-Alouds in Usability Testing: An International Survey. IEEE Trans. Prof. Communication, 55:2–19, 03 2012.
- [MH12] Martin, Bella; Hanington, Bruce: Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions. Rockport Publishers, hardcover edition, 2 2012.
- [No02] Norman, Donald A.: The design of everyday things. Basic Books, [New York], 2002.
- [Sc20] Schlott, Verena Eileen; Korkan, Ege; Kaebisch, Sebastian; Steinhorst, Sebastian: W-ADE: Timing Performance Benchmarking in Web of Things. In (Bielikova, Maria; Mikkonen, Tommi; Pautasso, Cesare, eds): Web Engineering. Springer International Publishing, Cham, pp. 70–86, 2020.