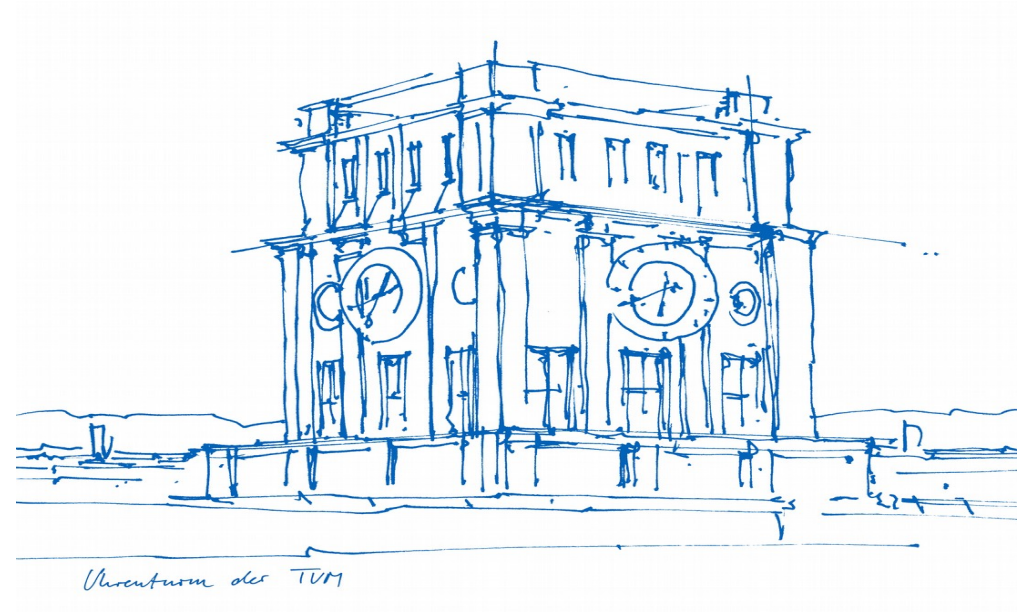


Testfest Results

Ege Korkan

Technical University of Munich



Agenda

- Feedback from implementation owners
 - What contradictions your implementation have with the test results
- Non tested assertions
- Parent-child assertions
- Assertions at risk : <2 implementations

Implementation Report

<https://github.com/w3c/wot-thing-description/blob/master/testing/report.html>

Make sure to download it and then open it from the browser as a regular html file.

Feedback from implementation owners

What contradictions your implementation have with the test results?

Take your result.csv files in outputs-tocheck, overwrite the results if you think they are false. Optionally, a comment would help us a lot to find difficult problems.

Feedback from implementation owners

The spreadsheets shown yesterday, which are just a csv files, lives here:

<https://github.com/w3c/wot/tree/master/testfest/2019-02-princeton/outputs-tocheck>

Example of Manual Sanity Check

There is an example overwrite as a PR: <https://github.com/w3c/wot/pull/684>

Example Manual Correction for SmartThings

No need to be merged but can be left until the end of the testfest



egekorkan committed 21 minutes ago

Verified

commit 03c78b6c1dbb150677781b99ae5c092916b7bf2d

2 testfest/2019-02-princeton/outputs-tocheck/SmartThings/impl-node-red.csv

Copy path

View file



	@@ -7,7 +7,7 @@
7	7 "server-data-schema-extras", "null",
8	8 "server-uri-template", "null",
9	9 "td-action-arrays", "not-impl",
10	- "td-action-arrays_forms", "not-impl",
10	+ "td-action-arrays_forms", "pass", "I have implemented this in tdmotionevent.jsonld",
11	11 "td-action-arrays_scopes", "not-impl",
12	12 "td-action-arrays_security", "not-impl",
13	13 "td-action-names", "not-impl",

Manual Sanity Check by Panasonic

<https://github.com/w3c/wot/pull/687/files>

Non tested assertions

Some assertions (27) CANNOT be tested automatically. Thus, we need manual input from implementation owners.

There is the template from Michael McCool who can be used, which is found here:

<https://github.com/w3c/wot/blob/master/testfest/2019-02-princeton/templates/manual.csv>

Filling this will automatically overwrite the null with your results

Non tested assertions (Examples)

- **Td-serialization-default-values:** For JSON-based Thing Description serialization, if values are not given for certain mandatory fields, default values **MUST** be assumed.
- **Td-form-protocolbindings:** Syntax If required, forms **MAY** be supplemented with protocol-specific vocabulary terms identified with a prefix.

And more related to behavior

Parent and Child Assertions - Discussion

Should a parent implementation if a child assertion about a non mandatory term is not implemented?

Example:

Td-action-names: Each optional vocabulary term as defined in the class Action and its superclass InteractionPattern MUST be serialized as a JSON name within an Action object.

In this assertion if there an implementation does not have, e.g. uriVariables in actions, this parent assertion is marked as not implemented.

- 1) Parent should pass: Since uriVariables are not mandatory, this doesn't mean that my implementation didn't implement td-action-names.
- 2) Parent should not pass: How can one know that you will implement uriVariables as a JSON name if you implemented terms.

Parent and Child Assertions : + and -

	Advantages	Disadvantages
1) Parent should pass:	Probably enough	Wishful Thinking
2) Parent should not pass	Rigorous	Rigorous, Can scare newcomers

I vote for option 2 since I find the option 1 as wishful thinking. We say that all action names, whether optional or not, must be JSON names.

At Risk Assertions

- Assertions with <2 implementations are marked as at risk at :
<https://github.com/w3c/wot-thing-description/blob/master/testing/inputs/atrisk.csv>

This also includes not tested assertions

Including child assertions, there are 106 assertions at risk!

At Risk Assertions

There are so many assertions that are implemented by no one OR they are implemented by only one implementation being node-wot with servients created specifically for testfest.

=> If there was a need to add all these features, where are the corresponding TDs?

E.g. There was a long discussion about unit but there is only one node-wot TD implementing that. (and this TD was built specifically for testfest)

At Risk Assertions

Simplifying:

- Scopes at interaction level gone => - 10 at-risk assertions
- Description, title like terms implemented in every possible level => - 12
- Data schema terms in properties and other interactions => -20
- Event cancellation, subscription => -6
- Manual assertions => -27
- Security schemes => -15

...