

What is the project about?

This project is in two phases.

The first one deals with implementing specialisations of the base HTML class in LibDOM, (the Netsurf Library for the Document Object Model Tree). The specific specialisations are known in advance and listed in the IDL definitions for Document Object Model. The goal is to successfully implement the missing classes, that pass the test suite. The primary goal, is to make libDOM faster, more compact and up to date.

The second phase provides two options of working either on LibHubbub or LibCSS.

In Libhubbub, the task revolves around identifying the differences between its own behaviour and the current html5 specifications. A decent effort will be made to put the library in line with the current specifications. Finding bugs in LibHubbub and passing the tests in the latest test suite is an intermediate goal. The technical details and end goals have been provided later on in this proposal.

In LibCSS, the task on hand is to select missing component modules and implementing it.

Personal Information

- Name : Rupinder Singh Khokhar
- IRC Nick : parwana
- Netsurf Wiki username : Parwana
- University : [International Institute of Technology \(IIIT\), Hyderabad](#)
- Field of study : Computer Science And Engineering
- Expected graduation date : 29 April, 2016
- Degree : Bachelor Of Technology In Computer Science And Engineering
- Email : rsk1coder99@gmail.com
- Telephone : +91-9810182623

About me:

I am an expert C programmer with about 5 years of experience. I am acquainted with an object oriented style revolving around structs. My interests lie in algorithms, data structures and problem solving. I regularly participate in Online programming contests on websites like Codeforces, Codechef, Topcoder and Spoj. And I can write compact C code with expertise. I have a fairly good understanding of DOM trees and have done few research projects involving trees in the past. I also have a good understanding of Netsurf's coding practices and ethics of the code and I am comfortable with the git workflow and repository management.. I am fairly conversant with LibHubbub's mechanism for tokenisation and parsing, since I have read the spec docs.

Programming languages

These are the programming/scripting languages/tools that I am fluent in:

- C, C++
- Bash, Javascript, ActionScript
- perl, python, ruby, octave

- HTML, CSS
- git

UNIX and C/C++ experience:

- Interval & segment trees (ResearchProject): [link](#)
- Built a client-server peer to peer communication program. It provides you an option to use either UDP or TCP: <https://bitbucket.org/rupindersingh/p2p-client-server>
- Made my own bash interpreting terminal in C: <https://bitbucket.org/rupindersingh/terminal-in-c>
- Project on semaphores and multi-threading: <https://bitbucket.org/rupindersingh/multithreading-and-semaphores>
- Built an AI Bot for playing the Backgammon Game: <https://bitbucket.org/rupindersingh/artificial-intelligence>
- 2D & 3D games in C++ opengl: <https://bitbucket.org/rupindersingh/opengl>

Team development:

I worked with a startup <http://www.greensoftware.in/> and made a health management system using Ruby On Rails. It is here:

<https://bitbucket.org/rupindersingh/health-management-system>.

My Coding Statistics:

Codeforces: <http://codeforces.com/profile/earthWalker>

Codechef: <http://www.codechef.com/users/rsk1994>

Apart from this, I have developed a few small games in Flash, available here: <http://web.iiit.ac.in/~rupindersingh.khokhar/>

Why I chose Netsurf?

In my endeavor to contribute towards FOSS, I stumbled across Netsurf. I liked the idea of working on the structure of a web browser. The codebase of Netsurf *Library* Code was properly structured and I could easily understand it. Reading Netsurf's codebase for the first time was a treat to my programmer's instinct. Thus this was my start of open source contribution. I hope to get more involved in it after I get selected for the Summer Of Code Program.

Abstract Planning:

Thorough discussions with the mentors about the possible solutions will take place, so that time is not wasted implementing something that ultimately is not suitable. A similar discussion took place between me and John Mark Bell for implementing the patch to improve entity support for LibHubbub.

If in any case mentors are busy or are unable to provide inputs to a specific Specialisation/tag, I will code a proof of concept and take a risk of having it thrown away. The POC would then be recursively improved to get the final result. If there is situation that I am severely stuck at a problem and the mentors are not responding at IRC, I will send a mail to the developer mailing list describing the problem, giving as much detail as possible. I will move onto some other problem in the meantime.

The work will proceed in several short iterations and regular commits. Any discrepancies will be corrected as per mentor's reviews. The focus would be more on LibHubbub. More time has been allocated for it because uncertain bugs/hurdles are bound to prop up.

Technical Planning:

- To avoid any dependency conflict, I have formulated the exact sequence for making the tag specialisations, which I endeavour to follow. The list has been put on Netsurf's wiki page: http://wiki.netsurf-browser.org/Sequence_planning_for_LibDOM_specialisations. In LibDOM, some tags are directly derived from the base class, while others make use of some other specialisation in the formulation of their attributes.
- Each specialisation/tag is expected to take only a very few days to handle, however it may vary based on the complexity of the class. Some involve just implementing the constructors, destructors and relevant get/set functions while others involve implementing specific behaviours as well. I already have implemented some tags and consider this task fairly easy to be accomplished.
- The testing for LibDOM would first be done on the existing test-suite. Own tests will be formulated in later stages.
- I have listed a few problems in LibHubbub and its possible solutions here on the netsurf wiki: <http://wiki.netsurf-browser.org/LibHubbub>. I will keep updating this list as more bugs and possible solutions are discovered. Once the list is complete, the data will be analysed. Some problems may have a combined solution. These will be discussed with the mentors so that time is not wasted in unnecessarily implementing solutions that are ultimately not suitable for the project. The latest spec docs available on the ideas page will be the best help.
- The specifications for LibHubbub are known before hand and are available on the ideas page. The sequence for implementation will be : Parser, followed by tokeniser, followed by tree-builder. This sequence will avoid any dependency conflict, as many functions of treebuilder are dependent on the tokeniser and that of tokeniser on the parser
- I have completed almost 50% work on LibHubbub already. The related branches are listed at the end of the proposal. The appropriate patches will be sent on the Mailing List soon.
-
- The testing for LibHubbub is done with latest html5lib testsuite.

Timeline for 1st Phase

A time of roughly one and a quarter month has been reserved for LibDOM. I would like the work to proceed in many small iterations. A rough deadline for each tag/specialisation implementation will be decided on between me and the mentor. Commits will be made as the work progresses. If I am unable to meet a short-term deadline, I will submit an explanation of the reason. However, the primary aim would be to stay ahead of schedule and put in extra work hours if not able to meet the deadline.

Week / Days	Work to be Done
22nd April - 2nd May (Pre-program preparation)	Prepare for start. Have discussions with mentors. Start with LibDOM
2nd May - 20th May (Week 1)	Implement tags in accordance with the sequence formulated, with short deadlines regular commits and reviews. Testing with the testsuite already provided.
20th May - 27th May (Week 2)	Continue implementing the missing functionalities in LibDOM --Evaluation period
28th May - 3rd June (Week 3)	Make own tests and stress test LibDOM. Any bugs encountered will be corrected too.
3rd June - 10th June (Week 4)	--Buffer Week 1

This timeline is however abstract and the main motive would be to wrap up the work as early as possible.

As stated in the description of the Phase 2, there are many small sub-projects that are not specified yet. Therefore, I cannot give an exact description of how the project will run. However, I can describe general characteristics of the project as I would like it to proceed:

(I would prefer working on LibHubbub.)

Generic Timeline for IInd Phase

10th June - 17th June (Week 5)	Identify and record the places where LibHubbub doesn't match with current html5 specs.
17th June - 24th June (Week 6)	Most of the generic problems are having a solution in the spec docs. For others, discussions with mentors of possible solutions to the problems and/or develop POC of solutions.
24th June - 1st July (Week 7)	Mid-Term Evaluations -Buffer Week 2
1st July - 8th July (Week 8)	Work on correcting the errors in LibHubbub according to the plan developed in previous stages.

8th July - 15th July (Week 9)	--do--
15th July - 22nd July (Week 10)	Buffer Week 4
22nd July - 29th July (Week 11)	Stress Test LibHubbub using alternate tests from other sources.
29th July - 5th August (Week 12)	Buffer Week 5
5th July - 12th August (Week 13)	Expected Pencil Down Date
12th July - 19th August (Week 14)	Creating Documentation for the work done.
19th July - 26th August (Week 15)	Firm Pencil Down & Final Evaluations

Note: buffer weeks are assigned to cover for pending work.

Other engagements in summer

I would be having no other demands on my time this summer. I will be available full time and work about 40 hrs per week. However, if any short plan of a holiday trip do come up, I will make it a point to inform the mentors well in advance.

Experience with Netsurf

LibDOM HR Tag(merged

patch):<http://git.netsurf-browser.org/libdom.git/commit/?id=2c5052d450577d5c26b902ae5f4402b87f76e406>

LibHubbub named Entities (pending mentor review) :

<https://bitbucket.org/rupindersingh/libhubub/commits/10ff264c57271431d635a154b19c6115e8714c7c>

Bitbucket Repository for LibHubbub work (patches will be sent soon):

Repo: libhubbub

sURL: <https://bitbucket.org/rupindersingh/libhubub.git>

sRef: mybranch3

latest SHA1: a56d0514b0c46fbcc546c3c228f724558cd47c2d

Land: master

Note: The work in this repo contains only POC's for removing the errors. Improvements will be decided after mentor reviews.

The following are the new (latest) tests that passed after the update:

tests1.test ; tests2.test ; tests3.test ; tests4.test ; entities.test ; numericEntities.test;
unicodeChars.test; namedentities.test; pendingSpecChanges.test

Bugs will be regularly updated on the wiki page mentioned in the technical planning section. Kindly have a look at them.