

Spineless Traversal for web layout

GANG ZHOU, College of William and Mary, USA

YAFENG WU, University of Virginia, USA

TING YAN, Eaton Innovation Center, Czech Republic

TIAN HE, University of Minnesota, USA

CHENGDU HUANG, JOHN A. STANKOVIC, and TAREK F. ABDELZAHER, University of Virginia, USA

Multifrequency media access control has been well understood in general wireless ad hoc networks, while in wireless sensor networks, researchers still focus on single frequency solutions. In wireless sensor networks, each device is typically equipped with a single radio transceiver and applications adopt much smaller packet sizes compared to those in general wireless ad hoc networks. Hence, the multifrequency MAC protocols proposed for general wireless ad hoc networks are not suitable for wireless sensor network applications, which we further demonstrate through our simulation experiments. In this article, we propose MMSN, which takes advantage of multifrequency availability while, at the same time, takes into consideration the restrictions of wireless sensor networks. Through extensive experiments, MMSN exhibits the prominent ability to utilize parallel transmissions among neighboring nodes.

CCS Concepts: • **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

Additional Key Words and Phrases: Wireless sensor networks, media access control, multi-channel, radio interference, time synchronization

ACM Reference Format:

Gang Zhou, Yafeng Wu, Ting Yan, Tian He, Chengdu Huang, John A. Stankovic, and Tarek F. Abdelzaher. 2010. Spineless Traversal for web layout. *ACM Trans. Web* 9, 4, Article 39 (March 2010), 3 pages. <https://doi.org/0000001.0000001>

1 TECHNICAL SECTION

1.1 Theory(4)

1.1.1 Attribute Grammar. We assumed that web page layout can be implemented in attribute grammar, and the attribute grammar have been scheduled into multiple 'there and back again' pass.

That is, web page layout call a sequence of mutating function, where each function:

This work is supported by the National Science Foundation, under grant CNS-0435060, grant CCR-0325197 and grant EN-CS-0329609.

Author's addresses: G. Zhou, Computer Science Department, College of William and Mary; Y. Wu and J. A. Stankovic, Computer Science Department, University of Virginia; T. Yan, Eaton Innovation Center; T. He, Computer Science Department, University of Minnesota; C. Huang, Google; T. F. Abdelzaher, (Current address) NASA Ames Research Center, Moffett Field, California 94035.

Authors' addresses: [Gang Zhou](#), College of William and Mary, 104 Jamestown Rd, Williamsburg, VA, 23185, USA; Yafeng Wu, University of Virginia, School of Engineering, Charlottesville, VA, 22903, USA; Ting Yan, Eaton Innovation Center, Prague, Czech Republic; Tian He, University of Minnesota, USA; Chengdu Huang; John A. Stankovic; Tarek F. Abdelzaher, University of Virginia, School of Engineering, Charlottesville, VA, 22903, USA.

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. Copyrights for components of this work owned by others than the author(s) 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.

© 2009 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1559-1131/2010/3-ART39 \$15.00

<https://doi.org/0000001.0000001>

- compute value for field x0, x1, x2...
- recursively invoke itself for each of the children, left to right.
- compute value for field y0, y1, y2...

Note that in the above formalization, both control flow and data flow is fixed and tied to the tree structure. We argue that this is a reasonable restriction to web layout as we are able to implement a wide variety of complex feature under such assumption.

We hypothesize that this is because web layout are local: the position of a dom node is highly related (i want to say determined but that sounds a bit too strong) to its relative. **Hard to write this section without referring to DSL**

1.1.2 Dirty Bit. The standard method for incremental web layout is the dirty bit algorithm. For each node, for each basic block BB, the algorithm keep a boolean variable, BB_dirtied, Similarly, for each procedure P, the algorithm keep a boolean variable, P_recursive_dirtied, both initially set at false.

A field is dirtied, when one of it's dependency change it's value. A basic block is dirtied when one of it's field is dirtied. A procedure P is recursively_dirtied, when one of it's corresponding basic block is dirtied, or when one of it's children's P is also recursively_dirtied.

To recalculate, the incremental algorithm traverse the tree for each procedure, there-and-back-again style, similar to how the value is initially calculated. However, only basic block with BB_dirtied=true will be re-executed, and once re-executed, said variable will be set to false. Additionally, once a node does not have the corresponding P_recursive_dirtied set, there is no need to recurse further.

1.2 PQ(5)

OM Init/Mark/Recalculate PQ Impl

1.3 Impl(3)

Below we give a syntax and semantics of the DSL megatron, and explain the design rational.

$M(\text{ain}) := P_N()$...

$P(\text{roc})_N := \text{self.BB_X}(); \text{children.P_N}(); \text{self.BB_Y}()$

$\text{BB}(\text{BasicBlock})_X := \text{self.N} <- T...$

$V(\text{ar}) := \text{unique symbols}$

$F(\text{unction}) := \text{a predefined set of primitive functions}$

$P(\text{ath}) := \text{self} \mid \text{prev} \mid \text{next} \mid \text{parent} \mid \text{first} \mid \text{last}$

$T(\text{erm}) := V \mid \text{if } T \text{ then } T \text{ else } T \mid F(T...) \mid P.V$

HTML Features

- visibility (display)
- position (static vs absolute)
- line breaking
- flex
- box model
- intrinsic width/height
- fixed width/height
- min/max width/height

We also have some features that is too small for a bullet point (e.g. image with width and not height/vice versa), which should not be talked about in detail, but i still think we should brief over. how should i structure it?

Received February 2007; revised March 2009; accepted June 2009