

Technical Disclosure Commons

Defensive Publications Series

April 21, 2017

Automatically Generating and Rendering Native Advertisements

Zhenguo Ou

Zekan Qian

Guannan Zhang

Tianjiu Yin

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Ou, Zhenguo; Qian, Zekan; Zhang, Guannan; and Yin, Tianjiu, "Automatically Generating and Rendering Native Advertisements", Technical Disclosure Commons, (April 21, 2017)
http://www.tdcommons.org/dpubs_series/475



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Automatically Generating and Rendering Native Advertisements

BACKGROUND

Native advertisement refers to an advertisement that follows the layout and functionality of a digital environment (e.g., a website or a mobile application) in which it is displayed. Unlike non-native advertisements (e.g., banner ads displayed on the top of a website) which can be disruptive and easily ignored by users, native advertisements are more effective and pleasant. This is because native ads can be seamlessly blended with original content displayed on the digital environment, and thus they are more likely to get users' visual attention and less likely to annoy users. As a result, many advertisement publishers want to implement native ads on their platforms in order to improve advertisement performance and user experience.

SUMMARY

This document describes technologies for automatically generating and rendering a native advertisement for display on a client device when the client device is accessing a publisher's webpage. A template node among a number of nodes of a webpage can be selected by the publisher. The selected template node specifies a content item that the publisher wants the native ad looks like. A template tag can be added to the template node to specify a set of templates suitable for rendering the native advertisement on a web browser. When a client device opens a web browser to access the publisher's webpage, the web browser can render the native ad by running a JavaScript downloaded from an advertisement server. The rendering process can include extracting a number of formatting attributes for the native ad from the template node, mapping the formatting attributes with a proper template from a number of

predefined templates maintained by an ad server, and loading the proper template from the ad server to render the content of the native advertisement in its corresponding inline frame.

DESCRIPTION OF DRAWINGS

Figure 1 shows an example system architecture for generating and rendering a native advertisement for display on a client device when the client device is accessing a publisher's webpage.

Figure 2 shows an example template node of a publisher's webpage in which a template tag is added to the template node.

DETAILED DESCRIPTION

FIG. 1 illustrates a system for automatically generating and rendering native ads for display on a client device when the client device is accessing a publisher's webpage. The system includes an advertisement server 106, a publisher 102, and a client device 104. The client device 104 can be a desktop computer, a mobile phone, or a tablet that is connected with a communication network such as the Internet. The advertisement server 106 maintains an advertisement database 114 that stores ads from advertisers (e.g., businesses). An advertisement can be in the form of a video, audio, image, text or a combination thereof. The advertisement server 106 also maintains a number of predefined templates 116 for ads. A template specifies, for example, layout, font, spacing, and headlines (i.e. the "look and feel") of an advertisement when rendered on a browser. The publisher 102 maintains a website that provides online content to audiences via the Internet. The publisher 102 can allow advertisements to be displayed on its website by joining an advertisement network 108 managed by the ad server 106. The ad display

network 108 allows advertisers (e.g., businesses) to reach a large number of audience through websites of publishers that join the network 108.

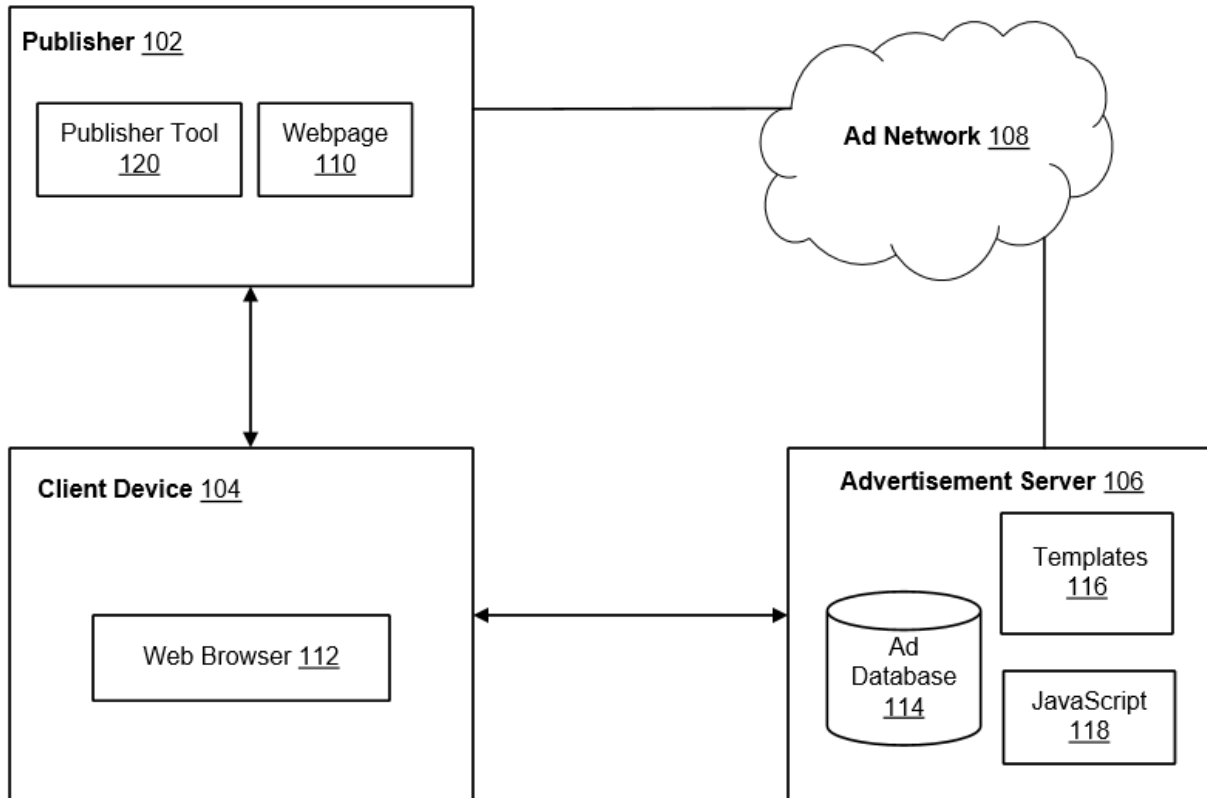


FIG. 1

The publisher 102 wants to generate a native advertisement that resembles the layout of original content displayed on the publisher’s webpage 110. A node from a number of nodes of the webpage can be selected by the publisher as a template node. The selected template node specifies a content item on the webpage that the publisher wants the native advertisement looks like. In some cases, the publisher 102 can be provided with a publisher tool 120 to select a template node and see a preview of the native advertisement on the publisher’s webpage.

A template tag can be added to the selected template node to specify a set of templates suitable for rendering the native advertisement on a web browser. An example template node with added template tag is shown in FIG. 2 below:

```

<ins class="adsbyadserver" style="display:inline-block;width:300px;height:250px;"
  data-ad-client="ca-pub-5620988105600723"
  data-native-ad-template=".article" ← Added template tag
  data-page-url="pp.com"></ins>
<script>
(adsbyadserver = window.adsbyadserver || []).push({});
</script>

```

FIG. 2

When a user of the client device 104 accesses the webpage of the publisher 102, a web browser 112 of the client device 104 loads the webpage from the publishers' server. The web browser 112 then sends a request to the ad server 106 to retrieve content of the native ad from the ad database 114. Subsequently, the web browser 112 runs a native ad JavaScript 118 (e.g., adsbyadserver.js) downloaded from the ad server 106 to start the rendering process. The JavaScript 118 first reads the template tag, which was added to the selected template node to identify the set of templates suitable for rendering the native advertisement. The JavaScript 118 then extracts a number of formatting attributes for the native ad from the template node. Examples of a formatting attribute include a size (e.g., width and height) of the template node, colors, font, font style (e.g., line height and font weight), and layout. The formatting attributes can be mapped with a proper template from a number of predefined templates 116 maintained by the ad server 106. The mapping process can be performed using one or more algorithms such as

a decision tree algorithm. The JavaScript then loads the proper template from the ad server 106 to render the content of the native advertisement in its corresponding inline frame. The JavaScript can automatically re-render the content of the native advertisement in its corresponding inline frame in response to a change in size of the browser or a change of the screen size of the client device.

ABSTRACT

Systems and methods for automatically generating and rendering a native advertisement for display on a client device when the client device is accessing a publisher's webpage are described. A template node among a number of nodes of a publisher's webpage can be selected to specify a content item that the publisher wants the native ad looks like. When the publisher's webpage is loaded by a web browser of a client device, the native ad can be rendered by the browser using a JavaScript downloaded from an advertisement server. The rendering process can include extracting a number of formatting attributes for the native ad from the template node, mapping the formatting attributes with a proper template from a number of predefined templates maintained by an ad server, and loading the proper template from the ad server to render the content of the native advertisement in its corresponding inline frame.