

SUFFOLK TECHNOLOGIES, LLC., v. AOL, INC & GOOGLE, INC, Appeal No. 2013-1392 (Fed Cir. May 27, 2014). Before Rader, Prost, and Chen. Appealed from E.D. Virginia (Judge Ellis).

### Background:

Suffolk sued Google and AOL for infringing a patent directed to computerized methods and systems for rendering a webpage based on a "received identification signal" (*i.e.*, referrer address). Google responded by arguing, among other things, that the asserted claims were anticipated by a newsgroup post. During claim construction the limitation "generating said supplied file" of claim 6 was construed by the district court to include "in dependence upon the received identification signal" from claim 1. Based on the claim construction, the district court granted Google's motion for summary judgment and held that claims 1, 7 and 9 were anticipated by the post, which qualified as "printed publication" prior art. Based on the district court's unfavorable claim construction of claim 6 and finding that the alleged prior art newsgroup post anticipated claims 1, 7 and 9, Suffolk stipulated to summary judgment of anticipation of claim 6 and appealed.

### Issues/Holdings:

Did the district court err by improperly construing the claim 6 limitation of "generating said supplied file" to depend upon the "received identification signal" from claim 1? No, affirmed. Did the district court err by granting summary judgment of anticipation of claims 1, 7 and 9 by the newsgroup post? No, affirmed.

### Discussion:

On appeal, Suffolk argued that: (i) the district court erred in construing the claim term (claim 6) because the generated file could depend on the "*content*" of the requesting web page and not just the received identification signal; and (ii) the district court improperly granted summary judgment of invalidity of claims 1, 7 and 9 over the newsgroup post.

Regarding claim construction, Suffolk argued that the claim language and the specification's implicit definitions and preferred embodiment do not support the district court's claim construction. The Federal Circuit determined that these arguments were unpersuasive because: (a) neither claims 1 nor 6 show any variables/mechanisms for conveying any information but the received identification signal, and (b) the excerpts of the specification relied upon, taken in context, support the claim construction of generating the correct file based on the received identification signal.

Regarding summary judgment, Suffolk argued that: (a) the post is not a "printed publication" because the audience was not those of ordinary skill in the art and locating the post would be too difficult, (b) small alterations to timestamps/email addresses (to avoid spams) could suggest to a jury that other inaccuracies existed, (c) expert testimony (which was different than the same expert's earlier testimony) was improperly excluded, and (d) genuine factual issues existed, even if their expert testimony is excluded. In finding Suffolk's argument unpersuasive, the Federal Circuit held that: (a) the post was a printed publication because those of ordinary skill did use the newsgroup, the technology was new, and the post was sufficiently disseminated to be, difficult or not, "publicly accessible" at the time of publication; (b) there was no evidence to suggest other inaccuracies existed in the post; (c) the district court did not abuse its discretion in finding intervening claim construction was not a sufficient reason to warrant a change in a validity opinion; and (d) there were no genuine issues of material fact based on expert testimony as Suffolk relied only on attorney arguments.

“How to tell which page called the script?” newsgroup posting (June 29, 1995)

"I am a newbie at this CGI [common gateway interface] stuff, so this question might seem ridiculous (I did look in the FAQ and on some web pagpages). I have this script that will be called from one of 18 pages. Depending on which page it was called from, the output will be different. Is there any environment variable that will tell me this, or do I have to externally pass information to the script."

A college student replied with the following post:

"Look at the CGI environment variable HTTP\_REFERER. In Perl, you can do something like this:

```
#!/usr/local/bin/perl
$referer = $ENV{'HTTP_REFERER'};
print "Content-type: text/plain", "\n\n";
if ($referer =~ /abc\.html/) {
    print "A link in abc.html called this document.",
        "\n"; }
elsif ($referer =~ /efg\.html/) {print "A link in efg.html called this document.", "\n"; }
else {print "A link in ", $referer, " called this document.", "\n"; }
```