

**IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF DELAWARE**

SHOPIFY INC. AND SHOPIFY (USA), INC.	)	
	)	
Plaintiffs and	)	
Counterclaim Defendants,	)	
	)	
v.	)	
	)	Civil Action No.1:19-cv-00439-RGA
EXPRESS MOBILE, INC.,	)	
	)	
Defendant and	)	
Counterclaim Plaintiff.	)	
	)	
	)	

**REBUTTAL EXPERT REPORT OF KEVIN C. ALMEROOTH, PH.D.**

By: Kevin C Almeroth

October 7, 2020

## TABLE OF CONTENTS

I.	INTRODUCTION AND SCOPE OF ENGAGEMENT .....	1
II.	BACKGROUND .....	2
	A. Background and Qualifications .....	2
III.	COMPENSATION .....	2
IV.	MATERIALS CONSIDERED .....	2
V.	LEVEL OF ORDINARY SKILL IN THE ART .....	3
VI.	UNDERSTANDING OF LEGAL PRINCIPLES.....	4
	A. Conception and Reduction to Practice.....	4
	B. Claim Construction.....	5
	C. Invalidity Based on Prior Art.....	6
	1. Prior Art.....	6
	2. Anticipation .....	9
	3. Obviousness.....	9
	4. Priority .....	13
	D. Written Description .....	13
	E. Enablement .....	13
	F. Claim Definiteness .....	14
	G. Patent Eligible Subject Matter.....	15
VII.	PRIORITY DATES OF THE PATENTS-IN-SUIT .....	17
	A. '397 and '168 Patents .....	17
	B. '755, '287, and '044 Patents.....	19
VIII.	SUMMARY OF INVALIDITY REBUTTAL OPINIONS.....	22
IX.	BACKGROUND AND TECHNOLOGY OF THE PATENTS-IN-SUIT .....	24
	A. Issues with Mr. Schmandt's Treatment of the State of the Art .....	25
	B. Issues with Mr. Schmandt's Overview of the Patents .....	30
	1. The '397 and '168 Patents .....	30
	2. The '755, '287, and '044 Patents.....	31
X.	ALLEGED PRIOR ART TO THE '397 AND '168 PATENTS .....	32
	A. Overview of The Asserted Prior Art References.....	32
	1. The Alleged SilverStream System.....	32
	a. Problems with the Scope of the Alleged "System" .....	32
	b. Description of SilverStream Version 1.5 .....	35
	2. Müller .....	40

(6) Element 37C(i) – “an external database containing data corresponding to said information stored in said internal database, and”	310
(7) Element 37C(ii) – one or more run time files.....	310
(8) Element 37C(iii) – where said run time files utilize information stored in said external database to generate virtual machine commands for the display of at least a portion of said one or more web pages .....	310
XII. THE FULL SCOPE OF THE ASSERTED CLAIMS OF THE ’397 AND ’168 PATENTS IS DESCRIBED .....	310
A. The Full Scope of the Asserted Claims of the ’397 Patent Is Sufficiently Described with Respect to the Term “Virtual Machine” .....	311
B. The Full Scope of the Asserted Claims of the ’168 Patent Is Sufficiently Described with Respect to the “Each Web Page Is Defined Entirely by Each of the Plurality of Objects Comprising that Web Page and the Style Associated with the Object” .....	313
XIII. THE FULL SCOPE OF THE ASSERTED CLAIMS OF THE ’397 PATENT ARE ENABLED WITH RESPECT TO THE “VIRTUAL MACHINE” CLAIM TERM .....	315
XIV. THE CLAIMED ELEMENTS OF THE ’397 AND ’168 PATENTS, BOTH INDIVIDUALLY AND AS AN ORDERED COMBINATION, WERE INVENTIVE IN 1999, AND WERE NOT CONVENTIONAL OR ROUTINE TO THE POSA .....	318
XV. ALLEGED PRIOR ART TO THE ’755, ’287, AND ’044 PATENTS .....	323
A. Overview of The Asserted Prior Art References.....	323
1. BlackBerry MDS Studio.....	323
2. Arner.....	326
3. Huang.....	327
4. Stevens.....	329
5. Rudolph .....	330
XVI. THE ALLEGED OBVIOUSNESS GROUNDS ASSERTED BY MR. SCHMANDT AGAINST THE ’755, ’287, AND ’044 PATENTS ARE MERITLESS .....	331
A. The Asserted Claims of the ’755, ’287, and ’044 Patents Are Not Obvious Over BlackBerry MDS Studio Alone .....	332
1. General Problems with Mr. Schmandt’s BlackBerry MDS Studio Obviousness Assertions and Grounds .....	332
2. Rebuttal of Mr. Schmandt’s Limitation-by-Limitation Analysis of the BlackBerry MDS Studio Alone Obviousness Ground.....	334
a. Element A – “A system for generating code to provide content on a display of a device, said system comprising”; and Element A’ – “A method of displaying content on a display of a device having a player and” .....	334
b. Element B – “computer memory storing a registry of” .....	335

c.	Elements C and C' – “a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C' – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and”	337
d.	Element D – “where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and”	340
e.	Element E – “b) the address of the web service;”	341
f.	Element F – “said method comprising:”	342
g.	Element G – “an authoring tool configured to:”	342
h.	Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service”	343
i.	Element I – “where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool”	346
j.	Element J – “access said computer memory to select the symbolic name corresponding to the web component of the defined UI object”	347
k.	Element K – “associate the selected symbolic name with the defined UI object”	348
l.	Element L – “where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name”	348
m.	Element M – “store information representative of said defined UI object and related settings in a database;”	349
n.	Element N – “retrieve said information representative of said one or more said UI object settings stored in said database; and”	350
o.	Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages”	350
p.	Element P – “produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and”	352
q.	Element Q – “produce a Player, where said Player is a device-dependent code;”	353
r.	Element R – “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object”	355

s.	Element S – “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service” .....	356
t.	Element T – “the web service utilizes the input symbolic name and the user provided one or more input values for generating .....	357
u.	Element U – “said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output values and provides instructions for a display of the device to present an output value in the defined UI object” .....	357
v.	Claim 13 of the '287 patent – “The system of claim 1, where said Player is activated and runs in a web browser” .....	358
w.	Claim 22 of the '755 Patent – “The method of claim 12, further comprising: providing said Application and Player over said network.” .....	360
x.	Claim 17 of the '044 Patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.” .....	360
y.	Claim 19 of the '044 Patent – “The method of claim 15, where said UI object is an input field for a web service” .....	360
B.	The Asserted Claims of the '755, '287, and '044 Patents Are Not Obvious Over Arner Alone or in Combination With Other References .....	361
1.	General Problems with Mr. Schmandt’s Arner Obviousness Assertions and Grounds .....	361
2.	Rebuttal of Mr. Schmandt’s Limitation-by-Limitation Analysis of the Arner Alone Obviousness Ground .....	361
a.	Element B – “computer memory storing a registry of” .....	361
b.	Elements C and C’ – “a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C’ – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and” .....	364
c.	Element D – “where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and” .....	365
d.	Element E – “b) the address of the web service;” .....	368
e.	Element F – “said method comprising:” .....	368
f.	Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service” .....	368
g.	Element I – “where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool” .....	370

h.	Element J – “access said computer memory to select the symbolic name corresponding to the web component of the defined UI object” .....	372
i.	Element K – “associate the selected symbolic name with the defined UI object”	372
j.	Element L – “where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name”	372
k.	Element M – “store information representative of said defined UI object and related settings in a database;” .....	375
l.	Element N – “retrieve said information representative of said one or more said UI object settings stored in said database; and” .....	376
m.	Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages” .....	377
n.	Element P – “produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and” .....	377
o.	Element Q – “produce a Player, where said Player is a device-dependent code;”	378
p.	Element R – “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object” .....	380
q.	Element S – “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service” .....	383
r.	Element T – “the web service utilizes the input symbolic name and the user provided one or more input values for generating .....	383
s.	Element U – “said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output values and provides instructions for a display of the device to present an output value in the defined UI object” .....	384
t.	Claim 13 of the ’287 patent – “The system of claim 1, where said Player is activated and runs in a web browser” .....	384
u.	Claim 22 of the ’755 Patent – “The method of claim 12, further comprising: providing said Application and Player over said network.” .....	385
v.	Claim 17 of the ’044 Patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.”	386
w.	Claim 19 of the ’044 Patent – “The method of claim 15, where said UI object is an input field for a web service” .....	386
2.	Rebuttal of Mr. Schmandt’s Arner and BlackBerry MDS Studio Obviousness Ground	387
a.	General Deficiencies in the Arner and BlackBerry MDS Studio Obviousness Ground .....	387

b.	Element L – where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name	389
c.	Element M – store information representative of said defined UI object and related settings in a database	390
d.	Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages”	391
e.	Element Q – “produce a Player, where said Player is a device-dependent code”	392
f.	Claim 17 of the ’044 patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow a video, or an RSS feed”	393
C.	The Asserted Claims of the ’755, ’287, and ’044 Patents Are Not Obvious Over Huang Alone or in Combination With Other References	394
1.	General Problems with Mr. Schmandt’s Huang Obviousness Assertions and Grounds	394
2.	Specific Issues with the Huang Alone Obviousness Ground	395
a.	Element A – “A system for generating code to provide content on a display of a device, said system comprising”; and Element A’ – “A method of displaying content on a display of a device having a player and”	395
b.	Element B – “computer memory storing a registry of”	396
c.	Elements C and C’ – “a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C’ – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and”	397
d.	Element D – “where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and”	398
e.	Element E – “b) the address of the web service;”	399
f.	Element F – “said method comprising:”	400
g.	Element G – “an authoring tool configured to:”	401
h.	Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service”	401
i.	Element I – “where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool”	402

j.	Element J – “access said computer memory to select the symbolic name corresponding to the web component of the defined UI object” .....	403
k.	Element K – “associate the selected symbolic name with the defined UI object”	403
l.	Element L – “where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name”	404
m.	Element M – “store information representative of said defined UI object and related settings in a database;” .....	404
n.	Element N – “retrieve said information representative of said one or more said UI object settings stored in said database; and” .....	406
o.	Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages” .....	407
p.	Element P – “produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and” .....	409
q.	Element Q – “produce a Player, where said Player is a device-dependent code;”	409
r.	Element R – “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object” .....	411
s.	Element S – “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service” .....	412
t.	Element T – “the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name,” .....	413
u.	Element U – “said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output values and provides instructions for a display of the device to present an output value in the defined UI object” .....	415
v.	Claim 13 of the ’287 patent – “The system of claim 1, where said Player is activated and runs in a web browser” .....	416
w.	Claim 22 of the ’755 Patent – “The method of claim 12, further comprising: providing said Application and Player over said network.” .....	417
x.	Claim 17 of the ’044 Patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.”	417
y.	Claim 19 of the ’044 Patent – “The method of claim 15, where said UI object is an input field for a web service” .....	418
3.	Specific Issues with the Huang and Rudolph Obviousness Ground .....	419
4.	Rebuttal of Mr. Schmandt’s Limitation-by-Limitation Analysis of the Huang and Rudolph Obviousness Ground .....	423



a.	Element A and A' – “A system for generating code to provide content on a display of a device, said system comprising”, and Element A' – “A method of displaying content on a display of a device having a player and” .....	423
b.	Elements B-C – “computer memory storing a registry of:”, Element C – “where the registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C' – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and” .....	424
c.	Element G – “an authoring tool configured to:” .....	426
d.	Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service,” .....	426
e.	Element Q – “produce a Player, where said Player is a device dependent code;” .....	427
f.	Element U – “3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.” .....	428
g.	Claim 17 of the '044 patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.” .....	429
h.	Claim 19 of the '044 patent – “The method of claim 15, where said UI object is an input field for a web service.” .....	430
D.	The Asserted Claims of the '755, '287, and '044 Patents Are Not Obvious Over Stevens in Combination With Other References .....	430
1.	Specific Issues with the Stevens and Rudolph Obviousness Ground.....	430
2.	Rebuttal of Mr. Schmandt's Limitation-by-Limitation Analysis of the Stevens and Rudolph Obviousness Ground .....	433
a.	Element A and A' – “A system for generating code to provide content on a display of a device, said system comprising”, and Element A' – “A method of displaying content on a display of a device having a player and” .....	433
b.	Element B – “computer memory storing a registry of:” .....	434
c.	Element C and C' – “a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service”, and Element C' – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and” .....	434

- d. Element D – “where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and” 435
- e. Element E – “b) the address of the web service;” ..... 436
- f. Element F – “said method comprising:” ..... 436
- g. Element G – “an authoring tool configured to:” ..... 437
- h. Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service,” ..... 438
- i. Element I – “where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool,” ..... 439
- j. Element J – “access said computer memory to select the symbolic name corresponding to the web component of the defined UI object,” ..... 439
- k. Element K – “associate the selected symbolic name with the defined UI object,” ..... 439
- l. Element L – “where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name,” ..... 440
- m. Element M – “store information representative of said defined UI object and related settings in a database;” ..... 440
- n. Element N – “retrieve said information representative of said one or more said UI object settings stored in said database; and” ..... 441
- o. Element O - “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages,” ..... 442
- p. Element P – “produce an Application including the selected symbolic name of the defined UI object, where said Application is a device independent code, and” ..... 443
- q. Element Q – “produce a Player, where said Player is a device-dependent code;” 444
- r. Element R – “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object,” ..... 445
- s. Element S – “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service,” ..... 446
- t. Element T – “2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name,” ..... 447
- u. Element U – “3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.” ..... 447

v.	Claim 13 of the '287 patent – “The system of claim 1, where said Player is activated and runs in a web browser.” .....	448
w.	Claim 22 of the '755 patent – “The method of claim 12, further comprising: providing said Application and Player over said network.” .....	449
x.	Claim 17 of the '044 patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.” .....	449
y.	Claim 19 of the '044 patent – “The method of claim 15, where said UI object is an input field for a web service” .....	450
3.	Huang and Stevens Obviousness Ground.....	452
a.	General Comments.....	453
b.	Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages,” .....	454
c.	Element Q – “produce a Player, where said Player is a device dependent code;” .....	455
d.	Claim 17 of the '044 patent – “The method of claim 15103, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.” .....	455
4.	Arner and Stevens Obviousness Ground.....	456
a.	General Comments.....	456
b.	Claim 13 of the '287 Patent – “The system of claim 1, where said Player is activated and runs in a web browser.” .....	457
c.	Claim 22 of the '755 Patent – “The method of claim 12, further comprising: providing said Application and Player over said network.” .....	458
d.	Claim 17 of the '044 patent – “The method of claim 14, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS fee” .....	458
XVII.	THE FULL SCOPE OF THE ASSERTED CLAIMS OF THE '755, '287, AND '044 PATENTS IS DESCRIBED.....	459
A.	The Full Scope of the Asserted Claims of the '755, '287, and '044 Patents Is Sufficiently Described with Respect to the Term “Player” .....	459
B.	The Full Scope of the Asserted Claim 1 of the '044 Patent Is Sufficiently Described with Respect to the “Authoring Tool Configured to . . . Build an Application Consisting of One or More Web Page Views From at Least a Portion of Said Database Utilizing at Least One Player” Claim Phrase.....	460
XVIII.	THE FULL SCOPE OF THE ASSERTED CLAIMS OF THE '755, '287, AND '044 PATENTS ARE ENABLED WITH RESPECT TO THE “PLAYER” CLAIM TERM ...	462
XIX.	THE ASSERTED CLAIMS OF THE '287 AND '044 PATENTS ARE NOT INDEFINITE .....	464
A.	Asserted Claims 1 and 13 of the '287 Patent and Asserted Claims 1, 17, and 19 of the '044 Patent Are Not Indefinite Based on the Claimed “Data Format Class Type” Associated With Each Symbolic Name .....	464

of the claim with “web services,” suggesting that a general compatibility with *web services* will motivate the POSA to incorporate specific *web components* into BlackBerry MDS Studio. In my opinion, Mr. Schmandt does not explain how this combination would work, or how the POSA would modify BlackBerry MDS Studio to integrate alleged functionality from Arner.

**C. The Asserted Claims of the '755, '287, and '044 Patents Are Not Obvious Over Huang Alone or in Combination With Other References**

**1. General Problems with Mr. Schmandt's Huang Obviousness Assertions and Grounds**

941. In Mr. Schmandt's Huang obviousness grounds, Mr. Schmandt treats the commonly used terms “application” and “player” that appear in Huang as automatically equivalent to the claimed application and player. Mr. Schmandt's explanations are generally deficient. In this ground in particular, Mr. Schmandt does not coherently explain why each limitation is allegedly taught within Huang. Typically, within each paragraph, Mr. Schmandt makes a single declarative statement, followed by a series of citations without meaningful explanation. This makes it difficult to assess what precisely Mr. Schmandt is arguing as the allegedly invalidating aspects of the prior art reference.

942. Additionally, Mr. Schmandt's conclusory statements typically suggest that Huang's use of particular web service standards---such as WSDL, SOAP, and UDDI---somehow imply that Huang necessarily invalidates the claims. However, in my opinion, these attempts blatantly mischaracterize the '755 patent, which extends beyond those pre-existing frameworks to create an architecture for efficient programming of web services, including abstractions for “web components” and “symbolic names,” among other things. (*See supra* Section IX.B.2; Almeroth Opening Report, Section V.D.). In my opinion, Mr. Schmandt's framing of this ground only underscores his oversimplification of the '755 patent family.

## 2. Specific Issues with the Huang Alone Obviousness Ground

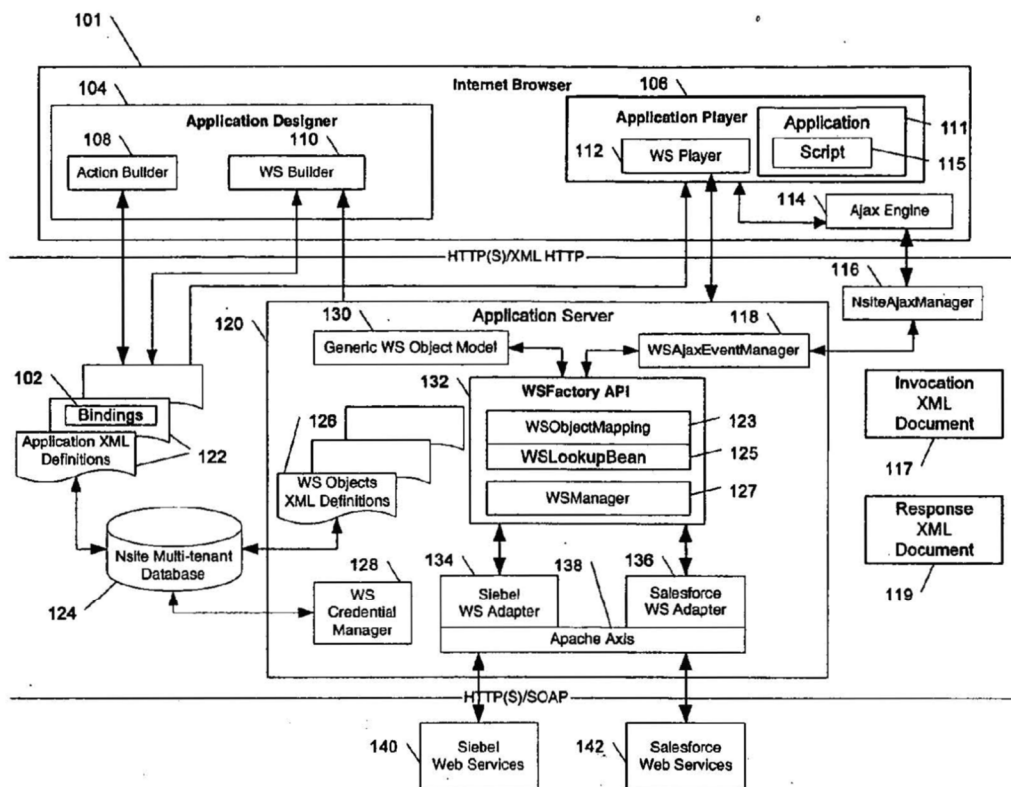
- a. **Element A – “A system for generating code to provide content on a display of a device, said system comprising”; and Element A’ – “A method of displaying content on a display of a device having a player and”**

943. With regard to “a system for generating code,” certain of the asserted claims require the generation of both application code and player code claims (e.g., claims 1, 12, and 22 of the ’755 patent). Mr. Schmandt does not explain how the Huang “Designer” generates any player code. In fact, the player code is provided along with the Designer to a user who wants to build applications: “[t]he Designer 104 and Player 106 are provided to the Internet Browser 101 by a server 120 as web pages.” (Huang at [0041]).

944. With regard to the “player” element recited in element A’, Mr. Schmandt fails to explain how the player of Huang meets the two requirements that he had argued to include in the construction of the claim player, and are included in the Court’s construction—the requirements that the Player contain device-specific code and be separate from the application. (*See* Schmandt Decl. I.S.O. C.C. Br., ¶ 220-227 (“one of ordinary skill in the art would recognize that the Player [sic] an executable file of device-specific instructions for the processor of the device that is generated by the authoring tool, is partitioned from the Application”)). The player of Huang is not device-dependent code that allows it to help run device independent applications across device platforms. In fact, neither the term nor the concept of device dependence appears anywhere in Huang or Nachnani.

945. As discussed in more detail below, Mr. Schmandt also does not explain how the player of Huang constitutes “device-specific code which contains instructions for a device and which is separate from the Application.” In fact, Huang describes an application as being “*in an*

Application Player.” (See, e.g., Huang at [0024], Fig. 1A). Therefore, Mr. Schmandt’s conclusory analysis of this element fails.



**FIG. 1A**

(SHOP0247755).

**b. Element B – “computer memory storing a registry of”**

946. Mr. Schmandt is deficient in his explanation of this limitation. He makes a single passing reference to “the database storing the WS Object XML definition” as the registry without any explanation. (See Schmandt Rep. ¶ 1199). As discussed above, in the context of the ’755 patent family, the registry is understood refers to a “web component registry” that includes “symbolic names” that associate the pertinent website elements (or web components) with a particular web service.

947. It is telling that neither the term “registry” nor the concept appears anywhere in Huang or Nachnani. According to Huang, the “WS Object XML definition” is “[t]he mapping between the Generic WS Object Model 130 and the vendor-specific adapters, such as the Siebel WS (Web Services) Adapter 134.” (Huang at [0044]). This is not the claimed registry. Therefore, Mr. Schmandt’s conclusory analysis fails for this reason alone.

- c. **Elements C and C’ – “a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C’ – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and”**

948. Mr. Schmandt again, for this limitation, provides a deficient explanation. As discussed above, the required symbolic names are bound to *web components* (i.e., elements of a web page), that are associated with inputs or outputs of web services.

949. Mr. Schmandt ignores the requirement of a relationship between symbolic names and web components. In the passage from Huang that Mr. Schmandt cites as evidence of a registry containing the claimed symbolic name, Huang makes clear that any alleged symbolic names in the “XML mapping file” (i.e., the purported registry) are not bound to web components but rather a generic class—the XML mapping file “defines a mapping between each attribute of a generic class such as the Account class shown above, and a class or classes in the specific web service vendor’s interface.” (See Schmandt Rep., ¶ 1202; Huang at [0081]). Mr. Schmandt fails to explain how the generic “Account” object constitute a web component or how the alleged symbolic name of “wsattributename=‘Name’ and wsattributename=‘LastModifiedDate’” evoke the web component. (See Schmandt Rep., ¶ 1203).

950. Setting aside for the moment that the objects that Mr. Schmandt identifies are not the claimed web services, none of the example objects that Mr. Schmandt cites to have both inputs and outputs. Mr. Schmandt concedes that “Huang does not explicitly state that each of its objects has both inputs and outputs.” (*See* Schmandt Rep., ¶ 1205). Mr. Schmandt’s conclusory remark aside, Mr. Schmandt fails to explain why a POSA would understand the specific objects in Huang has both inputs and outputs associated with a web service.

951. Mr. Schmandt relies extensively on the Salesforce “Opportunity” object as evidence for “a plurality of symbolic names of inputs and outputs associated with each web service” that are included in “each web component.” The attributes of the “Opportunity” object, an “WS Object,” are not inputs and output to a web service because WS Objects are created by the “WS Factory 120” to map “the generic WS Object interface 130 to a vendor-specific object 154” (Huang at [0057]).

952. Lastly, Mr. Schmandt attempts to improperly bootstrap the WSDL 1.1 specification into the Huang Alone Obviousness ground. The observation in Huang that WSDL is a common data format for web service interface descriptions does not automatically make example definitions of unrelated web services in the WSDL specification applicable to the specific WS objects in Huang or that generic web service definitions in WSDL teach evoking the claimed web components. They are not. (*See* Schmandt Rep., ¶ 1205; Huang at [0044]).

**d. Element D – “where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object, and”**

953. Mr. Schmandt’s arguments with respect to this limitation are again deficient. As a threshold matter, for the reasons I gave in my analysis of Element C and C’, Mr. Schmandt



does not articulate any symbolic name that is within the scope of the claims. For that reason alone, Huang does not teach this limitation, which has the requirement of a “symbolic name.”

954. With regard to the requirement that “each symbolic name has an associated data format class type,” Mr. Schmandt equates the claimed “data format type” to “component types.” (See Schmandt Rep., ¶ 1210). I disagree. The specification of the ’755 patent shows examples of “data format type” or “data type” as “Boolean,” “Int,” “String,” “multilineString,” among others. (See ’755 patent at 14:52-55, Table I). This is different from component type which is shown by Huang to be “Text Box,” “Text Area,” “Date Picker,” among others, in the table immediately following paragraph [0042] of Nachnani that Mr. Schmandt cites to. (See Schmandt Rep., ¶ 1210).

955. Moreover, Mr. Schmandt does not show how any of his alleged symbolic names has a preferred UI object. In fact, Mr. Schmandt concedes that “Huang does not explicitly disclose that the symbolic names of the object have a “preferred UI object”.” (See Schmandt Rep., ¶ 1212). The example Mr. Schmandt cites to, “IsClosed,” is not the claimed “symbolic name” that evokes web component but an attribute of a Web Service Object. (See prompt in Huang at Fig. 4C (“Map object attributes from the web service object to the controls from your NSite application.”)).

956. Lastly, I disagree with Mr. Schmandt’s suggestion in paragraph 1208 that that aspect of the claims render the claim indefinite for the reasons I discuss below in Section XIX.

**e. Element E – “b) the address of the web service;”**

957. Mr. Schmandt posits that “when Huang discloses that users provide a URL for a desired web service” Huang discloses Element E. (See Schmandt Rep., ¶ 1213). I disagree. First, Mr. Schmandt simply guesses that the URL shown in Figure 3 of Huang is provided by the

system and not manually entered by the user based on the phrase of Figure 3. (*See* Schmandt Rep., ¶ 1213). But looking at the specification describing the figure, it is at least equally likely that the URL is manually entered: “FIG. 3 is an illustrative drawing of defining a web service connection in an application Designer. A Connection screen 302 of a web services builder is shown, which allows a user to define parameters for communicating with a web service, including a vendor 304 of the web service, a URL 306 of the web service, and a user name 308 and a password 310 for logging in to the web service.” (Huang at [0105]). More importantly, Mr. Schmandt does not explain how a user provided URL is the claimed “address of the web service” found in a registry. In fact, Mr. Schmandt makes no showing that the URL he alleges to be the claimed web service address is found in the “WS Object XML definition” that he earlier alleged to be the claimed registry, which as I have explained in the preceding limitations is also incorrect.

**f. Element F – “said method comprising:”**

958. This method, according to this limitation, must be one of the following: “[a] method of displaying content on a display of a device utilizing a registry of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and where the registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, and b) the address of the web service,” “[a] method of displaying content on a display of a device having a Player, where said Player is a device dependent code,” or “[a] method of displaying content on a display

of a device having a player and non-volatile computer memory storing symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and where each symbolic name has a preferred UI object, and an address of the web service.” All of the asserted method claims of the ’755, ’287, and ’044 patents share the common goal of providing a cross-platform way to consistently display information by using a combination of player, application, symbolic names, web components, and web services.

959. In my opinion, Huang is not directed at allowing users to displaying content on a display of a device having a player, where said player is a device dependent code, or directed at utilizing a registry of one or more web components related to inputs and outputs of a web service obtainable over a network. Instead, as discussed above, Huang is about accessing databases and/or business logic of other organizations via web services and to do so in a “vendor-independent” manner.

**g. Element G – “an authoring tool configured to:”**

960. As discussed above, for the claims that require an authoring tool to produce a player, Mr. Schmandt fails to show the alleged authoring tool (i.e., “Designer UI”) generates any player code. Moreover, as discussed in connection other limitations, Mr. Schmandt fails to show the alleged authoring tool is configured to select the required symbolic names. (*See* Schmandt Rep., ¶ 1216).

**h. Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds**

**to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service”**

961. The Designer UI in Huang does not define UI objects that “correspond[] to the web component included in said registry” because, as discussed above, Huang does not have the required registry within the scope of the claims. In fact, Mr. Schmandt makes no mention of the “XML mapping file” that he alleges to be the claimed registry. (*See* Schmandt Rep., ¶¶ 1218-1220).

962. As discussed above, in the ’755 patent, the claimed registry provides functionality that allows a user to generate UI objects based on the bindings between web components and web services that are built into the system. Mr. Schmandt does not identify any analogous or similar functionality in what he asserts to be the authoring tool within Huang. Moreover, as previously discussed, the attributes of WS Objects in Huang are not “an input of the web service and an output of the web service.”

- i. **Element I – “where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool”**

963. For this element, Mr. Schmandt fails to show “UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool.” Instead, Mr. Schmandt attempts to incorporate disclosure of Nachnani and posits that Nachnani’s description of placing a UI component by a user into a workspace meets this limitation. (*See* Schmandt Rep., ¶¶ 1221-1222). I disagree.

964. Simply placing an UI component or an object within a visual editor does not meet this claim limitation because according to the claim the UI object has to be a preferred UI object

that corresponds to the symbolic name of the web component. Here, Mr. Schmandt makes no mention of any symbolic name or any web component in connection with his discussion of the alleged UI component. Nor has Mr. Schmandt pointed to a registry as claimed in the '755 patent.

**j. Element J – “access said computer memory to select the symbolic name corresponding to the web component of the defined UI object”**

965. Mr. Schmandt, in this limitation, does not address the location of the computer memory that is used to select the symbolic name corresponding to the web component of the defined UI object. (*See* Schmandt Rep., ¶¶ 1223-1225). Fundamentally, Huang does not have a registry or any symbolic name to select within a registry.

966. Moreover, Mr. Schmandt does not identify any symbolic name or any web component the symbolic name corresponds to. For example, in the Data Source Mappings screen 402 that Mr. Schmandt cites to, a user is presented with a Vendor Name (i.e., Salesforce) and asked to select a Web Service Object. As previously discussed, Mr. Schmandt does not explain whether he is alleging a Web Service Object in Huang to be an UI object, a web component, or a web service. (*See* Schmandt Rep., ¶ 1224). In any case, Web Service Objects (WS Objects) in Huang is an abstract software component that is not an UI object, not a web component, and not a web service.

**k. Element K – “associate the selected symbolic name with the defined UI object”**

967. Mr. Schmandt’s asserts that this limitation is satisfied by Huang’s description of “bindings between application components and web service objects and their attributes.” (*See* Schmandt Rep., ¶ 1226). But Mr. Schmandt fails to identify any symbolic name selected from computer memory.

968. Whether the bindings in Huang can be represented as “data in an Extensible Markup Language (XML) based format” is irrelevant to whether the selected symbolic name “corresponding to the web component of the defined UI object” exists, as the claims require. (See Schmandt Rep., ¶ 1227).

**l. Element L – “where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name”**

969. Again, as with Element K, Mr. Schmandt fails to identify any symbolic name selected from computer memory that corresponds to the web component of the defined UI object. (See Schmandt Rep., ¶ 1228). The figures in Huang that Mr. Schmandt cites to appear to show attributes of WS Objects for a user to select. This is fundamentally different from symbolic names corresponding to the web component of the defined UI object.

970. Therefore, Mr. Schmandt’s arguments on this limitation are inapposite.

**m. Element M – “store information representative of said defined UI object and related settings in a database;”**

971. For this element, Mr. Schmandt cites to three figures of database schemas in Nachnani (i.e., Figs. 6c, 13, and 14 shown below) as evidence of storing UI objects in a database. (See Schmandt Rep., ¶ 1229). Looking at the three figures, it is evident that none of them show information representative of any defined UI objects being stored in a database. Figure 6c, for example, shows content information from an application (name and date info), Figures 13 and 14 show information about company processes and dates.

Figure 6c

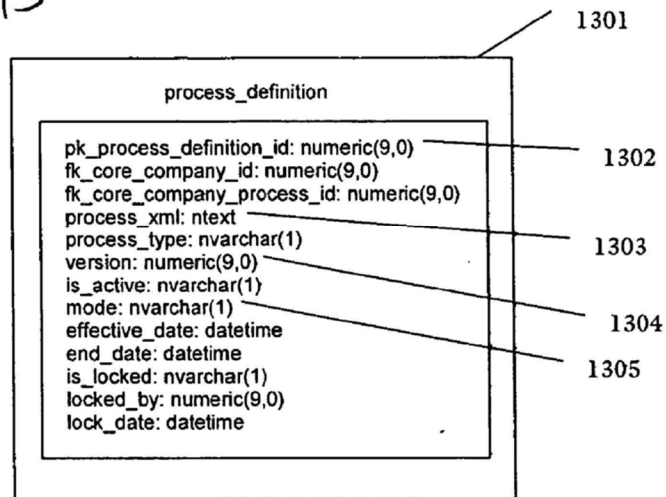
Example Application Data in Database

id	dtc1	dtc2	sc1	sc2
1	4/22/2005		John J. Smith	

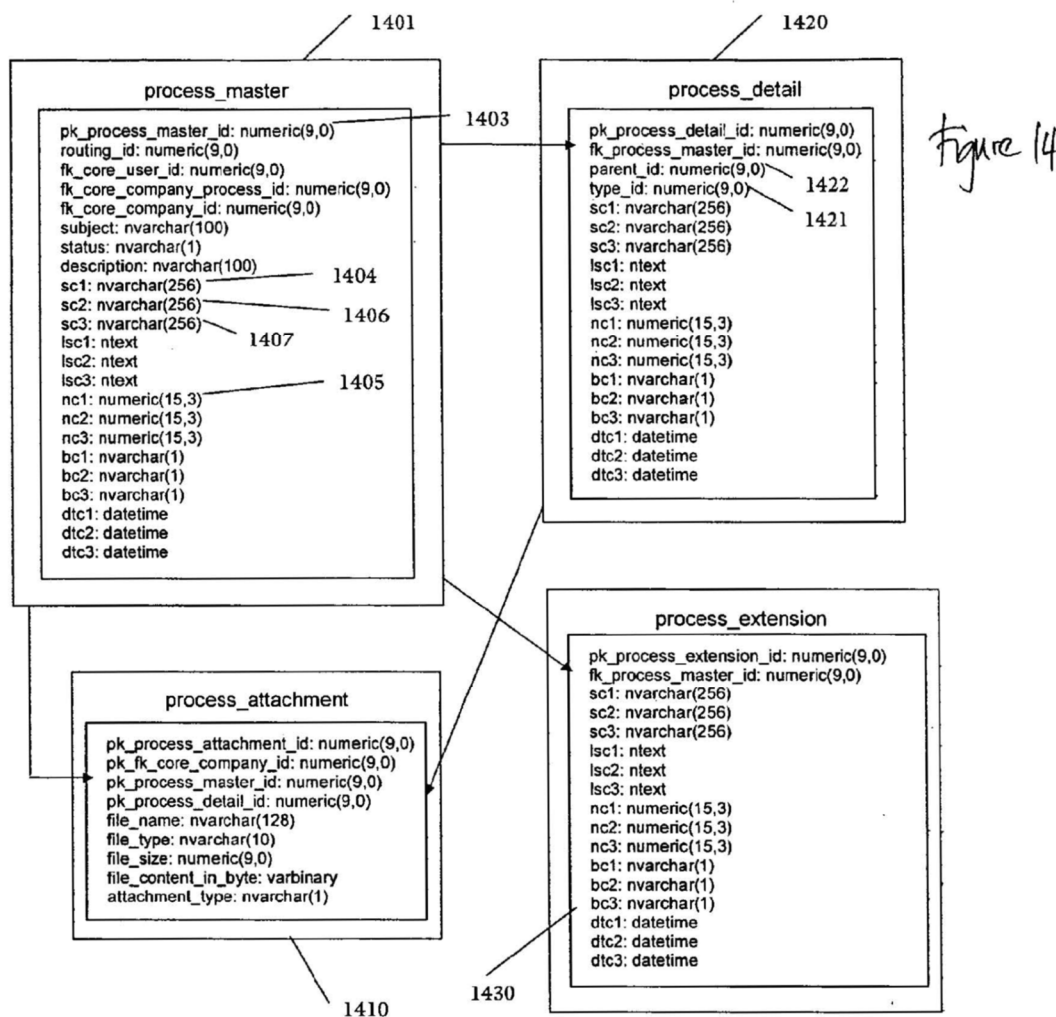
Figure 6c is a table with 5 columns and 3 rows. The columns are labeled 'id', 'dtc1', 'dtc2', 'sc1', and 'sc2'. The first row contains the values '1', '4/22/2005', an empty cell, 'John J. Smith', and an empty cell. The second row contains five empty cells. Reference numerals 612, 614, 616, 618, 620, 622, and 624 point to various elements of the table: 612 points to the first column, 614 to the 'id' header, 616 to the 'dtc1' header, 618 to the 'sc1' header, 620 to the value '1', 622 to the value '4/22/2005', and 624 to the value 'John J. Smith'.

(SHOP0247808).

figure 13



(SHOP0247824).



(SHOP0247825).

- n. **Element N – “retrieve said information representative of said one or more said UI object settings stored in said database; and”**

972. For this claim element, I incorporate my analysis from Element M. Without having stored information representative of the defined UI objects in a database, there is nothing to be retrieved. Moreover, the passages from Huang that Mr. Schmandt cites to actually show the application data being loaded is “data concerting [sic] a particular person or transaction” and not information representative of the defined UI objects. (See Schmandt Rep., ¶ 1231; Huang at [0108] (“A user 118 of the Player UI 114 may invoke a command to load a previously-saved



application, including application data, such as data concerning a particular person or transaction, data from the database 110 into the Player UI 114.”); id. at [0097] (“A user 118 of the Player UI 114 may invoke a command to load a previously-saved application, including application data, such as data concerning [sic] a particular person or transaction, data from the database 110 into the Player UI 114.”).

- o. Element O – “build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages”**

973. To the extent that Mr. Schmandt suggests that aspects of this claim element are indefinite in paragraph 1233, I disagree for the reasons I provide below in Section XIX.

974. For this claim element, I incorporate my analyses from Elements M and N above. Once again, this element requires the aforementioned database. Specifically, it requires Huang to include a player that “utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages.” Mr. Schmandt has not identified any such database, and in fact confirms my analysis in limitations M and N. For this reason alone, Huang does not teach this claim element. (*See* Schmandt Rep., ¶¶ 1232-1240).

975. Moreover, the Huang application is not the claimed application for at least three additional reasons. First, as discussed above in Element A, Mr. Schmandt does not explain how the player of Huang is “separate from the Application.” In fact, Huang describes an application as being “in an Application Player.” (*See, e.g.*, Huang at [0024], [0118], [0120], [0122], Fig. 1A). Huang also describes application components as a part of the player: “Then the WS Player 112 populates the application components of the Player 106 according to the bindings 102.” (Huang at [0124]). Moreover, as Mr. Schmandt points out in his discussion of the next element,

the alleged application includes JavaScript, the same language that the alleged player is implemented in. (*See* Schmandt Rep., ¶¶ 1242).

976. Second, the Huang application does not run entirely on the user device. The claims require that “the Application and Player are provided to the device and executed on the device.” (*See, e.g.*, ’755 patent at 37:31-32). Huang, much like Arner above, is about building server-based applications that can run in a web browser but have its business logic handled by an intermediate server or using an application server as an intermediary between what is running on a browser and an actual web service. As shown in Figure 8 of Nachnani, to the extent that Application Definition 838 is what Mr. Schmandt alleges to be the claimed application, although Application Definition 838 is on the Client Computer 831, the actual application is on Application Server 850.

977. Third, Mr. Schmandt fails to show that Huang builds “an application consisting of one or more web page views.” Mr. Schmandt makes no attempt to show the “Application Definition 108” that he cites to as the alleged application consists of any web page views. To the extent that Mr. Schmandt cites to a player presenting “‘web page views’ to the user,” that occurs after the building of the application. (*See* Schmandt Rep., ¶¶ 1238-1240).

978. Lastly, Mr. Schmandt attempts to improperly combine features from BlackBerry MDS and Arner to modify the Huang Designer which is premised on at least having the Designer to include a “test” mode. (*See* Schmandt Rep., ¶ 1234). In an attempt to overcome these deficiencies, in paragraph 1234, Mr. Schmandt appears to attempt to convert this single-reference obviousness ground into a multiple reference obviousness ground. Mr. Schmandt concedes that the Huang Designer does not have a “test” mode, but simply states that “POSA would understand that the Designer could incorporate the player’s “test” mode.” (*See* Schmandt

Rep., ¶ 1234). I understand that analyzing prior art is not about opining what could have been disclosed but what is actually disclosed. Although I disagree that having a test mode in the Huang Designer means a POSA would be motivated to combine BlackBerry MDS and Arner with Huang, but without it, a POSA is much more unlikely to be motivated to do so.

**p. Element P – “produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code, and”**

979. For this claim element, I incorporate my analysis from Element O. With respect to this limitation, Mr. Schmandt asserts that the Applications created by the Designer 104, including the XML definitions and user-defined script 106 such as JavaScript, is the claimed application. (*See* Schmandt Report, ¶ 1242). But he does not identify the full scope of the alleged Application and what it includes. As discussed in Element O, Mr. Schmandt fails to explain how alleged application is separate from the alleged player consistent with the requirement of the asserted claims.

980. Mr. Schmandt does not explain why, in his opinion, the alleged application is “device-independent code” especially when he has identified JavaScript that is a part of the alleged application is device-dependent code when he later discusses the player in Element Q. (*See* Schmandt Report, ¶ 1247).

981. Lastly, as discussed above, Mr. Schmandt does not identify or explain which symbolic name of the defined UI object is included in the alleged application. In fact, the terms “symbolic name” and “the defined UI object” do not appear anywhere in his discussion of this element.

**q. Element Q – “produce a Player, where said Player is a device-dependent code;”**

982. For this limitation, I incorporate my analysis from Elements O and P above. In my opinion, it is plain that Huang does not “produce a Player, where said Player is a device-dependent code.”

983. Mr. Schmandt concedes that with respect to claim 1 of the ’755 patent, “the Designer does not ‘produce’ the player” and that “Huang does not disclose how the player program is created.” (*See* Schmandt Rep., ¶ 1246). But he then makes the conclusory statement that a POSA would have been motivated to modify Huang to have the Designer to produce a player. I disagree.

984. The Huang Designer 104 is specifically an “Application Designer” and is referred to as such throughout the specification. (*See, e.g.*, Huang at Figs. 1A, 2A, 2B, 7; [0014], [0015], [0020], [0067], [0078], [0103]-[0105], [0114], [0118], [0133]). More importantly, Huang states that “Applications created by the Designer 104 are stored in Extensible Markup Language (XML) format as XML definitions 122 in a database 124, and loaded into the Player 106 as the application 111.” (Huang at [0042]). Saving data in XML format is fundamentally different from and significantly simpler than generating Java Server Pages (JSP) or JavaScript that implements a player. It is an entirely different and much more difficult undertaking. Neither the idea of using Designer 104 to generate JSP/JavaScript nor the how to do so is disclosed anywhere in Huang and Nachnani. Mr. Schmandt does not explain how this wholesale change can be made to Huang Designer 104 and why such change would not break existing functionality of Designer. In my opinion, a POSA would not be motivated to create a whole new designer that can simultaneously generate XML definitions for an application and JSP/JavaScript for a player.

985. Moreover, Mr. Schmandt mischaracterizes Huang as disclosing the Designer as “producing JavaScript as part of the application.” (*See* Schmandt Rep., ¶ 1247). Huang only

describes associating an application component with an existing script. (*See* Huang at [0057] (“The application component 170 may be associated with a script 196, which is part of the application.”). Moreover, the JavaScript has to be programmed by the user and is not automatically generated by the Designer. (*See* Huang at [0057] “The script 196 includes user-defined computer program code in a programming language such[]as JavaScript<sup>TM</sup> or the like.”).

986. Lastly, Mr. Schmandt argues that because JavaScript can be device-dependent code that any code written in JavaScript is automatically the claimed player of device-dependent code. I disagree. The device dependent-code in the claimed player are device dependent not because the language they are written in but the function the code performs. The device-dependent code in the claimed player is intended for, among other things, “adapting the Application to the resources and limitations of any particular device. Some of these areas of adaptation include the speed of the device’s microprocessor, the presence of device resources such as cameras and touch screens.” (’755 patent at 34:52-56). These device-dependent functionalities have to be specifically programmed and do not come automatically by the virtue of using a language such as JavaScript. Mr. Schmandt does not point to where in Huang disclosure of implementing such device-dependent functionalities can be found, because they are not there.

**r. Element R – “such that, when the Application and Player are provided to the device and executed on the device, and when a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object”**

987. For this element, I incorporate my analyses of limitations O, P, and Q from above. As discussed above, Huang is about building server-based applications that can be run in a web browser but having all of its business logic handled by a server. As shown in Figure 8 of Nachnani, while the Player Client 833 is on the Client Computer 831, the App Player JSP 860 is also on Application Server 850. Similarly, to the extent that Application Definition 838 is what

Mr. Schmandt alleges to be the claimed application, while Application Definition 838 is on the Client Computer 831, the actual application is on Application Server 850.

988. Moreover, Mr. Schmandt provides no explanation how “a user of the device provides one or more input values associated with an input symbolic name to an input of defined UI object,” an input value is associated with an input symbolic name, or which “symbolic name” and “defined UI object” he is referring to. (*See* Schmandt Rep., ¶¶ 1248-1249).

**s. Element S – “1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service”**

989. For this element, I incorporate my analysis of Element R immediately above. Mr. Schmandt cites a series of passages from Huang and Nachnani without any analysis. The passages he cites to do not necessarily support the conclusion that this limitation is met by Huang.

990. For example, Mr. Schmandt cites to a passage from Huang that describes Figure 1D (shown below). (*See* Schmandt Rep., ¶ 1250). To the extent that any user provided input values and corresponding input symbolic names are provided by a device, Figure 1D actually shows that they are provided to a Server WS Factory and not to a web service as required by the claim. Similarly, in paragraph 74 of Nachnani that Mr. Schmandt cites, it generally describes “[t]he Player UI executes actions and evaluates formulas in response to events that are associated with user interface components such as buttons and tables” but says nothing about providing “the user provided one or more input values and corresponding input symbolic name to the web service.” (*See* Schmandt Rep., ¶ 1250).

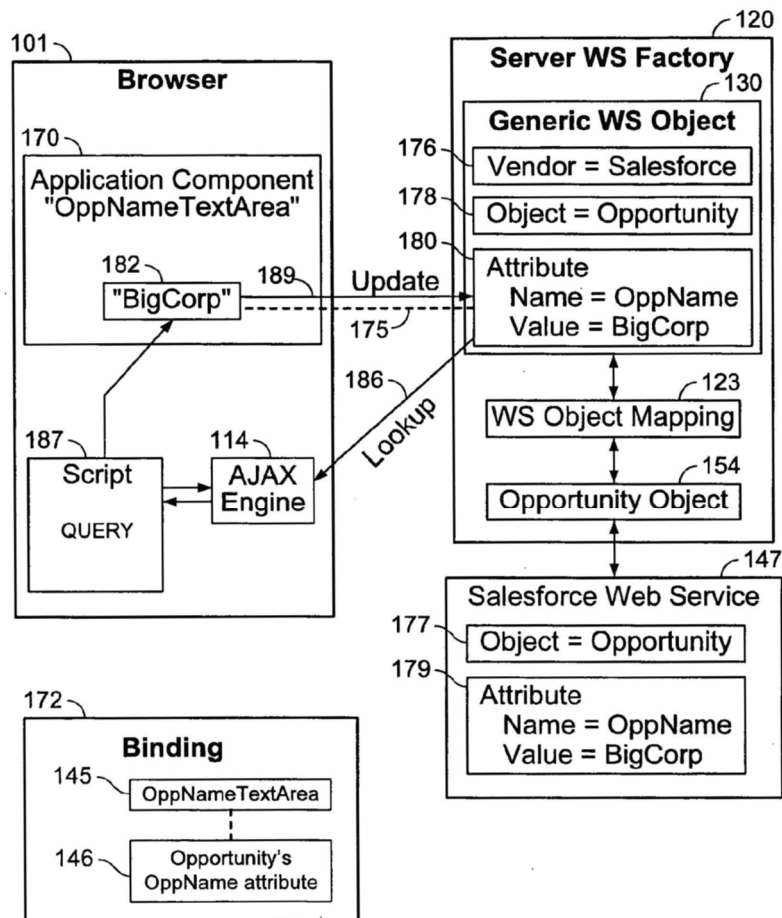


FIG. 1D

(SHOP0247758).

991. Additionally, to the extent that any user input values are provided to a web service, Mr. Schmandt does not explain how the “corresponding input symbolic name” associated with the defined UI objects in those examples are furnished to the web service.

- t. **Element T – “the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name,”**

992. For this element, I incorporate my analysis of Elements R and S above. Once again, Mr. Schmandt cites a series of disjointed passages from Huang and Nachnani without any analysis or explanation. Most of the cites passages are generic description of the functionality of WSDL and SOAP. For example, Mr. Schmandt cites to Huang at paragraph [0004] which

merely offers a generic description of web service and WSDL—“More precisely a web service is an application component that communicates via open protocols, processes XML messages, describes the messages using XML schema, and provides an endpoint description using WSDL. WSDL is the Web Service Description Language. WSDL makes it possible to describe an endpoint for a message and its behavior. WSDL layers additional information over the XML schema definitions that describe actual messages.” (*See* Schmandt Rep., ¶ 1251).

993. Moreover, Mr. Schmandt does not identify an actual web service, an input symbolic name, or user provided input values, let alone explain how a web service “utilizes the input symbolic name and the user provided one or more input values.” In fact, the cited passages show that the web service is utilizing attributes from Vendor-Specific WS Objects for processing rather than utilizing “the input symbolic name and the user provided one or more input values.” (*See* Schmandt Rep., ¶ 1251; Huang at [0066], [0104], Fig. 1B). In fact, in Figure 1B that is being described in the cited paragraphs [0066] and [0067] of Huang, it shows that Vendor Web Service 156 is processing a Vendor Specific WS-Object 154 which is created by the Server WS Factory 120, and is not utilizing “the input symbolic name and the user provided one or more input values.”



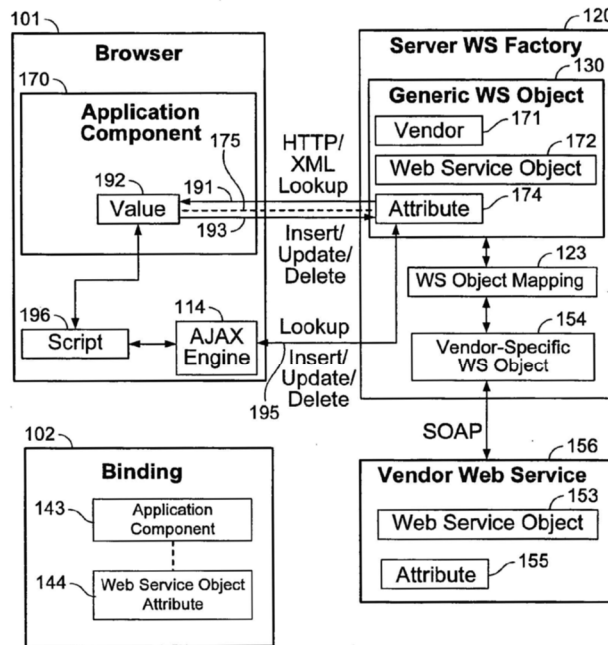


FIG. 1B

(SHOP0247756).

- u. **Element U – “said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output values and provides instructions for a display of the device to present an output value in the defined UI object”**

994. For this element, I incorporate my analyses of elements R, S, and T above.

Again, Mr. Schmandt cites a series of passages from Huang and Nachnani without any analysis or explanation. Mr. Schmandt does not provide an explanation of how the Player receives the output symbolic names and output values and uses that to generate an output.

995. In actuality, as shown in Figure 1B above, which is described by paragraph [0067] cited by Mr. Schmandt, Huang does not depict any player receiving any response from outside of the Browser 101. Instead, it shows Application Component 170 interacting directly using Script 196 which has been identified by Huang and by Mr. Schmandt as a part of the

application. (*See* Schmandt Rep., ¶ 1216; Huang at [0058] (“The application component 170 may be associated with a script 196, which is part of the application.”)).

996. Moreover, as depicted in Figure 1B and described in Huang, what is received by the Application Component 170 is Generic WS Object 130, and not the output symbolic name and corresponding one or more output values of a web service.

997. Lastly, none of Mr. Schmandt’s cited passages show a player providing “instructions for a display of the device to present an output value in the defined UI object.” For example, in paragraph [0054] of Huang, the Player 106 allows applications executing in the Player 106 to interact with web services by “inserting, updating, or deleting a value in a web service object.” In paragraph [0040], the Player 106 “receive values from web services, send values to web services, and invoke other operations provided by web services.” And in paragraph [0067], application player 106 “insert[s] the application component’s value into the web service attribute.” None of these examples and the rest of the cited passages say anything about a player providing “instructions for a display of the device to present an output value in the defined UI object.”

**v. Claim 13 of the '287 patent – “The system of claim 1, where said Player is activated and runs in a web browser”**

998. I incorporate my arguments above that plainly establish that the Huang does not teach the claimed player. Moreover, with respect to this claim element, moreover, as discussed above, at least the portion of the alleged player, (e.g., the App Player JSP 860) is running on a server (e.g., Application Server 850).

999. I also note that the second of the two passages Mr. Schmandt cites in this section only describes running an application program in a web browser and make no mention of any player. (*See* Schmandt Rep., ¶ 1253; Nachnani at [0019] (“In general, in a seventh aspect, the

invention features a method of running an application program in a web browser, including the acts of providing at least one client computer and one server computer in communication with the network, executing a web browser on the client computer, executing a client portion of the application program on the web browser, wherein the client portion presents a runtime user interface according to an application definition.”)).

**w. Claim 22 of the '755 Patent – “The method of claim 12, further comprising: providing said Application and Player over said network.”**

1000. For this limitation, I incorporate my analyses in Elements O, P, Q, and R. As I have discussed extensively above, the alleged player in Huang is a static application. It is not provided to a device as a result of actions taken in the authoring tool, or during the processes that are contemplated within the systems and methods of the '755 patent.

1001. I also note that in paragraph [0016] of Nachnani cited by Mr. Schmandt, the web browser is transmitting information to a server to create a software application and says nothing about providing an application and player over the network. (*See* Schmandt Rep., ¶ 1254; Nachnani at [0016] (“Furthermore, the web browser may transmit the defined aspects to a server over a computer network to create the software application.”)).

**x. Claim 17 of the '044 Patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.”**

1002. Mr. Schmandt does not cite to any disclosure in in Huang that actually discloses the various types of web components recited in this claim limitation. Instead, Mr. Schmandt makes some general reference to WSDL that is irrelevant to the claim limitation at issue. Mr. Schmandt then cites to paragraph [0042] of Nachnani to suggest that “the application components that can be bound to web services include ‘image.’” (*See* Schmandt Rep., ¶ 1255).

But that paragraph does not describe any image web component but merely an image “user interface element.” (Nachnani at [0042]).

1003. Lastly, in the event that the image user interface element were found not to disclose “a text chat, a video chat, an image, a slideshow, a video, or an RSS feed,” Mr. Schmandt simply states that “such web services were well known as of the priority date for the ’755, ’287, and ’044 patent.” (*See* Schmandt Rep., ¶ 1255). As an initial matter, this claim limitation requires that “web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed” and not web service of these varieties. Mr. Schmandt further asserts, without any explanation or support, that because “Huang discloses that a ‘text box’ consisting of a ‘text input field’” it “confirms that a “text chat” web service could easily be used with Huang’s system.” (*See* Schmandt Rep., ¶ 1255). I disagree. Having a text box in no way means a text chat can be easily implemented. To implement a text chat requires coding numerous functionalities unique to a chat application above and beyond having a text box.

1004. Lastly, Mr. Schmandt cites to Steven and Arner because, for example, Stevens allegedly discusses a “chat” service. (*See* Schmandt Rep., ¶ 1256). Whether these other references discuss a “chat” service, is irrelevant to whether Huang discloses this limitation. If Mr. Schmandt were attempting to combine Huang with the other two references, he provides no explanation regarding why a POSA would be motivated to so modify Huang and how such modification would be accomplished.

1005. Therefore, it is my opinion that Mr. Schmandt has failed to demonstrate that claim 17 would have been obvious.

y. **Claim 19 of the ’044 Patent – “The method of claim 15, where said UI object is an input field for a web service”**

1006. Mr. Schmandt cites to several passages without any explanation on how these passages support the conclusion that Huang meets this claim limitation. Huang does not. For example, Mr. Schmandt cites to paragraph [0103] of Huang, but that passage only shows that “the application component 206 is a text input component” but not that it is an input field for a web service. (*See* Schmandt Rep., ¶ 1257). Similarly, in cited paragraph [0054], Huang describes “application user interface components, such as text fields, for receiving data values from a user” and nothing about an input field for a web service. Again, in cited paragraph [0067], Huang describes “[a] user of the application player 106 can lookup values for the application component (e.g. the text area), from the associated web service attribute, as well as insert the application component’s value into the web service attribute . . .” but says nothing about input for a web service. As discussed above, the web service attributes are attributes of a WS Object and not input fields for a web service.

### **3. Specific Issues with the Huang and Rudolph Obviousness Ground**

1007. With regard to Huang and Rudolph, there are a number of general deficiencies that Mr. Schmandt ignores. In my opinion, the POSA would not have been motivated to combine Huang with Rudolph because the two are targeted at different audiences and use different technologies that run crosswise with each other.

1008. For example, Huang is targeted at enterprises that want “to access databases and/or business logic of other organizations via web services.” (Huang at [0007]). In the first paragraph of the Summary of Huang, it describes an example of applications used to access the web services are “software applications (i.e., programs) which carry out a business process.” (Huang at [0006]). The only example web services used in Huang are “Siebel Web Service 140

provided by Oracle Corporation of Redwood Shores, Calif., and the Salesforce Web Service 142 provided by Salesforce.com of San Francisco, Calif.” (Huang at [0043]).

1009. As such, Huang is for software developers who work for enterprises who can write the necessary scripts to make the Huang’s system work. (Huang at [0052] (“However, such client invocations are useful because they allow a user to write scripts in a language such as JavaScript™ to customize the behavior of applications. The Web Services AJAX API allows users to write such scripts to customize application behavior by interacting with web services.”)). For example, to access a web service such the Salesforce web service after a user presses button, a user has to program a script. (See Huang at [0052] (“a user-defined script that is called when a button component is pressed can call the WSFactory API 132 to interact with web services, such as the Siebel web service 140 and the Salesforce web service 142.”)).

1010. In contrast, and as Mr. Schmandt recognizes, “Rudolph relates to website creation tools “targeted to the consumer market” to “allow end users to create websites and other types of media . . . with little or no knowledge of web languages.” (See Schmandt Rep., ¶ 1260; Rudolph at 1:10-13).

1011. Moreover, Huang is a client-server-based system where the client portion runs in a browser and interfaces with Java Server Pages (JSP) or Java Server Faces (JSF) pages on a server, while Rudolph is a desktop application that runs locally on a personal computer and outside a browser. To be clear, Rudolph uses WebKit to execute its web widgets during development. But WebKit is not a browser and in fact it is for developers who want to display certain web content in their own desktop applications and outside a browser. (See <https://developer.apple.com/documentation/webkit#overview> (“Call WebKit functions and methods only from your app’s main thread or main dispatch queue.”)). Therefore, the Rudolph

system itself is a desktop application as shown in the figure below and described in Rudolph. The website creation tool in Rudolph (i.e., “iWeb® website creation tool”) is a desktop application that runs in the application layer (Rudolph at 5:29-32 (“the application layer can include iWeb Core and Web Widget applications. The iWeb Core code includes code for the iWeb® website creation tool and the web widget code includes code for implementing web widgets.”)).

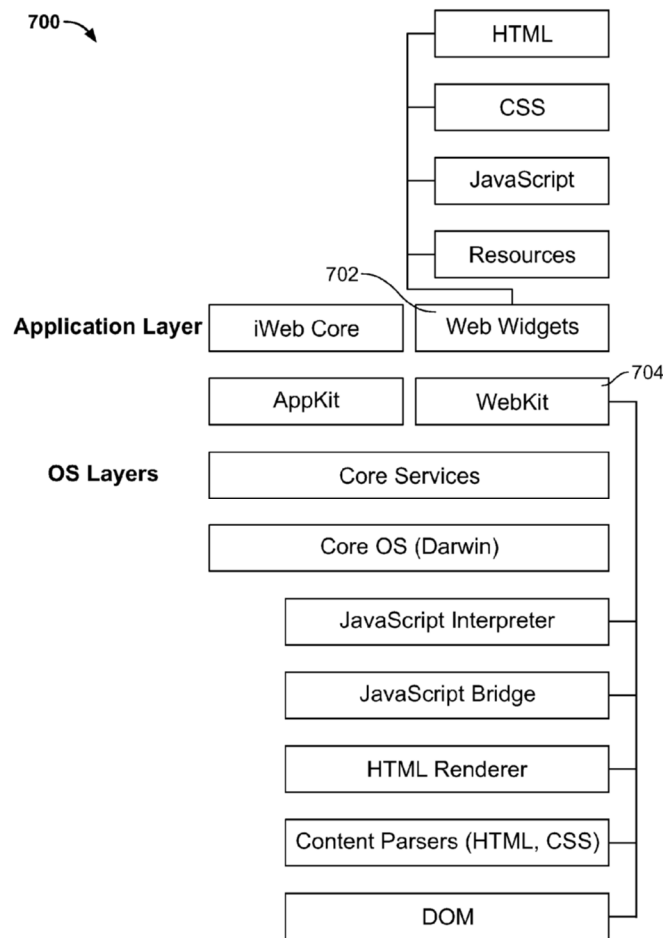


FIG. 7

(SHOP0011631).

1012. Moreover, Apple’s press release further confirms that iWeb, the website creation tool described in Rudolph, is a desktop application. (*See*

<https://web.archive.org/web/20100307021303/http://www.apple.com/pr/library/2006/jan/10ilife.html>)

1013. Mr. Schmandt does not account for these differences in arguing that there is an alleged motivation to combine Huang with Rudolph, or that there would have been a reasonable expectation of success in doing so. Moreover, Mr. Schmandt does not articulate a viable motivation to combine the Huang with Rudolph beyond a statement that “[b]oth references were programs and applications used to create web applications,” and that “[i]n general, computer programs can be adapted to include additional features by programming those new features.” (Schmandt Report, ¶ 1261). This is clearly erroneous. Under this theory, every piece of software, regardless of the software architecture, programming language, or technology used can be combined so long as each piece of software is in the same field of use.

1014. Beyond the failure to articulate a specific motivation to combine or reasonable expectation of success, a central flaw with Mr. Schmandt’s Huang and Rudolph obviousness ground is that it does not account for the claimed application and player, and their attendant features (such a device-dependence, device-independence, separation, etc.), required by the inventions of the ’755, ’287, and ’044 patents.

1015. Finally, Mr. Schmandt does not articulate any argument pertaining to reasonable expectation of success. Fundamentally, Mr. Schmandt’s argument is flawed because the proposed combinations, in some instances, would require the conversion of an entire desktop program written in a desktop-application programming language (e.g., C, Objective-C, etc.) into a client-server-based program in JavaScript and Java Server Pages. Mr. Schmandt provides no explanation as to why or how this is possible, or how doing so would not undermine the core functionality of the Huang.



**4. Rebuttal of Mr. Schmandt's Limitation-by-Limitation Analysis of the Huang and Rudolph Obviousness Ground**

- a. Element A and A' – “A system for generating code to provide content on a display of a device, said system comprising”, and Element A' – “A method of displaying content on a display of a device having a player and”**

1016. Mr. Schmandt, for this limitation, provides very little explanation and the explanation he does offer is deficient. As discussed above, for the limitation of “a system for generating code,” certain of the asserted claims require the generation of both application code and player code. Mr. Schmandt does not explain how the website creation tool generates the “WebKit,” the alleged player. The Rudolph website creation tool plainly does not. The passage that Mr. Schmandt cites to states that “WebKit® is an application framework included with Mac OS X which is the basis of Mac OS X's Safari web browser.” (*See* Schmandt Rep., ¶ 1265; Rudolph at 5:44-46).

1017. While Mr. Schmandt cites to the general reservation language in Rudolph that its system can be implemented in a computing system that includes back end, middleware, or front end components, Rudolph does not disclose how that implementation would be carried out, and Mr. Schmandt is also silent on that subject. (*See* Schmandt Rep., ¶ 1264).

1018. The only purported motivation to combine that Mr. Schmandt offers is the general assertion that the teachings of the references are “directed to the same field.” (*See* Schmandt Rep., ¶ 1264). As discussed above, even if the two references were directed at the same field, they are targeting two very different audiences: enterprise software developers in Huang versus consumers with no programming knowledge in Rudolph. The black box approach of Rudolph's web widgets would not be helpful to the developer users of Huang because the developers expect and need to make customization.

1019. Moreover, components of the desktop-application-based website creation tool in Rudolph cannot be easily grafted onto Huang without a wholesale rewrite in different languages (e.g., JavaScript and Java Server Pages) and for a different software architecture (e.g., client-server). And a POSA would have little expectation of success without undue experimentation.

1020. Lastly, I disagree that a POSA would be motivated to add rich media types discussed in Rudolph to Huang because Huang is for building business applications that have very little need for the kind of rich media types discussed in Rudolph.

- b. Elements B-C – “computer memory storing a registry of:”, Element C – “where the registry includes: a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service,” and Element C’ – “of one or more web components related to inputs and outputs of a web service obtainable over a network, where each web component includes a plurality of symbolic names of inputs and outputs associated with each web service, and”**

1021. I incorporate my discussion above regarding how these elements are not met in Huang. For these elements, Mr. Schmandt bases his analysis on the faulty premise that the web widgets in Rudolph is equivalent to the claimed web components. They are not.

1022. Mr. Schmandt does not identify any “symbolic names of inputs and outputs associated with each web service” included in the web widgets of Rudolph. Mr. Schmandt does not identify any symbolic names required for evoking the web widgets if they were the claimed web components. Mr. Schmandt does not explain how the web widgets could interact with the claimed application and player in a manner consistent with the claimed invention. Mr. Schmandt does explain, if symbolic names were used, or how these symbolic names required for evoking web widgets would be stored in a registry.

1023. Mr. Schmandt asserts that “web widgets disclosed in Rudolph could be stored as WS Object XML definitions in the database as both WS Object XML definitions.” (*See* Schmandt Rep., ¶ 1264). It is unclear what Mr. Schmandt is trying to prove with this statement. But Mr. Schmandt fails to explain how JavaScript, CSS, and other code that is used to implement Rudolph web widgets can be stored in the XML format of Huang.

1024. Mr. Schmandt further fails to explain how the Rudolph web widgets can be executed by the Huang player. They cannot. Mr. Schmandt also fails to explain how the WebKit, which executes Rudolph web widgets, that Mr. Schmandt alleges to be the player could be modified to execute XML definitions rather than JavaScript, CSS, and other code.

1025. Mr. Schmandt also asserts that “web services widgets disclosed by Rudolph could be integrated into the database of Huang where applications are stored.” (*See* Schmandt Rep., ¶ 1269). Even if that were true, Mr. Schmandt fails to explain how the WebKit can execute JavaScript, CSS, and other code stored in Huang’s databases without additional modification or what modification is required to make WebKit capable of executing code stored in Huang’s databases.

1026. Lastly, Mr. Schmandt argues that a POSA would be motivated to combine Rudolph and Huang because Huang teaches “Rapid Application Development (RAD) tools.” (*See* Schmandt Rep., ¶ 1270). But this is a reason that would actually discourage a POSA from combining Rudolph with Huang. A Rapid Application Development tool is still tool for developing software applications. Huang is particularly focused on developing business applications. On the other hand, Rudolph, as its Summary states, is “[a] media authoring tool includes web widgets for creating page elements for various types of media (e.g., websites, blogs, broadcasts, slide shows, multimedia presentations).” (Rudolph at 1:30-33). In my opinion,

a POSA would find the AppKit/WebKit techniques that Rudolph teaches to be inapplicable to the client-server and XML-based Huang system.

**c. Element G – “an authoring tool configured to:”**

1027. As discussed above, the media authoring tool of Rudolph does not produce player code as required by some of the asserted claims. Moreover, the media authoring tool of Rudolph is a desktop application written on top of AppKit desktop application framework. As previously discussed, components of the desktop-application-based website creation tool in Rudolph cannot be easily added to Huang without a wholesale rewrite in different languages and for a different software architecture. And a POSA would not be motivated to do so and would have little expectation of success without undue experimentation.

**d. Element H – “define a user interface (UI) object for presentation on the display, where said UI object corresponds to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service,”**

1028. Mr. Schmandt’s explanation for this element is deficient because, as discussed above, Mr. Schmandt fails to show that the web widgets in Rudolph are equivalent to the claimed web components. Again, they are not.

1029. Even assuming the web widgets were the claimed web components, Mr. Schmandt fails to show how the web widget manipulation functionalities of Rudolph’s desktop-application-based website creation tool can be grafted on to the Huang Designer that is programmed to handle XML definitions.

1030. Moreover, as discussed above, components of the desktop-application-based website creation tool in Rudolph cannot be easily added to Huang without a wholesale rewrite. And a POSA would have little expectation of success without undue experimentation, especially

not for any purported incremental benefit to have “web components that are already associated with the necessary UI objects.” (*See* Schmandt Rep., ¶ 1276).

**e. Element Q – “produce a Player, where said Player is a device dependent code;”**

1031. For this element, Mr. Schmandt appears to argue that the web widgets of Rudolph constitute the claimed player. (*See* Schmandt Rep., ¶ 1277).

1032. This contradicts his prior position that the WebKit is the alleged player. (*See* Schmandt Rep., ¶ 1265 (“Specifically, Rudolph discloses a player in the form of Safari’s WebKit.”)). This also contradicts Mr. Schmandt’s prior position that the web widget is the claimed web component. (*See* Schmandt Rep., ¶ 1267 (“In addition, Rudolph discloses that a web component that could be added to a web application is a web widget.”)).

1033. Mr. Schmandt cites to the passage in Huang that its web widgets “can differentiate whether it is running in a media authoring tool or in a browser (e.g., Safari®)” as support for the proposition that the web widgets contain the claimed device dependent code. (*See* Schmandt Rep., ¶ 1277; Huang at 2:56-57). I disagree because the alleged code that differentiates whether it is running in a media authoring tool or in a browser is still being executed by WebKit, the same device platform. (*See* Rudolph at Fig. 7, 5:43-46 (“the web widget code in the application layer uses WebKit® services. WebKit® is an application framework included with Mac OS X which is the basis of Mac OS X's Safari web browser.”)). Moreover, this code is not the type of device-dependent code in the claimed player that is separate from the device independent application, and is intended for, among other things, “adapting the Application to the resources and limitations of any particular device. Some of these areas of adaptation include the speed of the device’s microprocessor, the presence of device resources such as cameras and touch screens.” (’755 patent at 34:52-56).

1034. In paragraph 1278, Mr. Schmandt cites to another reference, Stevens, for the proposition that a POSA would be motivated to include code that allows an application to distinguish between operation in the designer module or the player module. Setting aside the issue of impropriety of adding another reference to this obviousness ground, it is unclear how this is relevant to the claim limitation at issue. In fact, allowing the application to distinguish the operating environment is the opposite of the claimed invention where the application is device-independent code that does not need to be aware of the different operating environment. Moreover, Mr. Schmandt does not explain how including “this functionality in the widget itself” would be implemented in the Huang Designer to make “the mechanisms for editing widgets would be built into the widgets themselves”. (*See* Schmandt Rep., ¶ 1278). In fact, contrary to Mr. Schmandt’s claim, supporting the web widgets of Rudolph in Huang, even if possible, would make the implementation of the Huang Designer much more complex, not less.

1035. Lastly, it is unclear what Mr. Schmandt is arguing in paragraph 1279 of his report, or why he believes that Express Mobile’s understanding of device-dependent code would lead a POSA to think code for WebKit is “non-functional in a web browser that is not based on WebKit,” or how this is relevant to the claim limitation at issue. (*See* Schmandt Rep., ¶ 1279).

1036. In any case, as I have discussed in detail above, a POSA would not be motivated to combine the teaching of Rudolph with Huang, especially not the alleged player in Rudolph.

**f. Element U – “(3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for a display of the device to present an output value in the defined UI object.”**

1037. I incorporate my discussion of Element Q immediately above. For this element, Mr. Schmandt again appears to argue that the web widgets of Rudolph constitute the claimed player. (*See* Schmandt Rep., ¶ 1281). As discussed above, this contradicts his prior position that

the WebKit is the alleged player as well as his prior position that the web widget is the claimed web component.

1038. On the other hand, if Mr. Schmandt maintains his allegation that the WebKit is alleged player in Rudolph, then that position is contradicted by the very passage that he cites for this claim limitation. Because as the cited passage shows, the web widget, not the alleged player, “can provide all the supporting code (e.g. HTML, JavaScript®, CSS) for connecting to the map service, sending requests for map data, receiving the map data in response to the requests, and the formatting of map data for presentation in the web widget 406.” (*See* Schmandt Rep., ¶ 1267; Rudolph at 4:25-31). This means that the steps performed by the claimed player are all done by, and from within, the alleged web components (i.e., web widgets).

1039. Lastly, I note that Mr. Schmandt again asserts that “Rudolph discloses to a POSA code included in the web widget that distinguishes the operating environment.” (*See* Schmandt Rep., ¶ 1280). To the extent that this statement is meant to imply that the alleged code in the web widget is the claimed device dependent code, I disagree. As discussed above, the alleged code does not perform the require functions of the claimed device dependent code and is not separate to device independent application.

**g. Claim 17 of the '044 patent – “The method of claim 15, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed.”**

1040. As discussed above, a POSA would not be motivated to attempt to transplant the web widgets of Rudolph into Huang because of the extensive software rewrite required. Also, as discussed above, Mr. Schmandt fails to show that the web widget of Rudolph is equivalent to the claimed web component. For example, Mr. Schmandt does not explain what are the “symbolic name of the web component,” what are the “non-volatile computer memory” that stores the

“symbolic name of the web component,” among others. Moreover, Mr. Schmandt fails to explain how the Rudolph web widgets would interact with the alleged player of Huang. In fact, Mr. Schmandt does not even explain how any of the Rudolph web widgets would interact with WebKit, the alleged player, consistent with the requirements of the claim.

**h. Claim 19 of the '044 patent – “The method of claim 15, where said UI object is an input field for a web service.”**

1041. As discussed above, a POSA would not be motivated to attempt to transplant the web widgets of Rudolph into Huang because of the extensive software rewrite required. Moreover, Mr. Schmandt does not identify any UI object included in the widgets of Rudolph that meets all the requirements of the claim such as the “input symbolic name to an input of the defined UI object,” the “symbolic name of the web component,” the “non-volatile computer memory” that stores the “symbolic name of the web component,” among others. Moreover, Mr. Schmandt fails to explain how any alleged UI object in the Rudolph web widgets would interact with the alleged player of Huang. In fact, Mr. Schmandt does not even explain how any alleged UI object in the Rudolph web widgets would interact with WebKit consistent with the requirements of the claim.

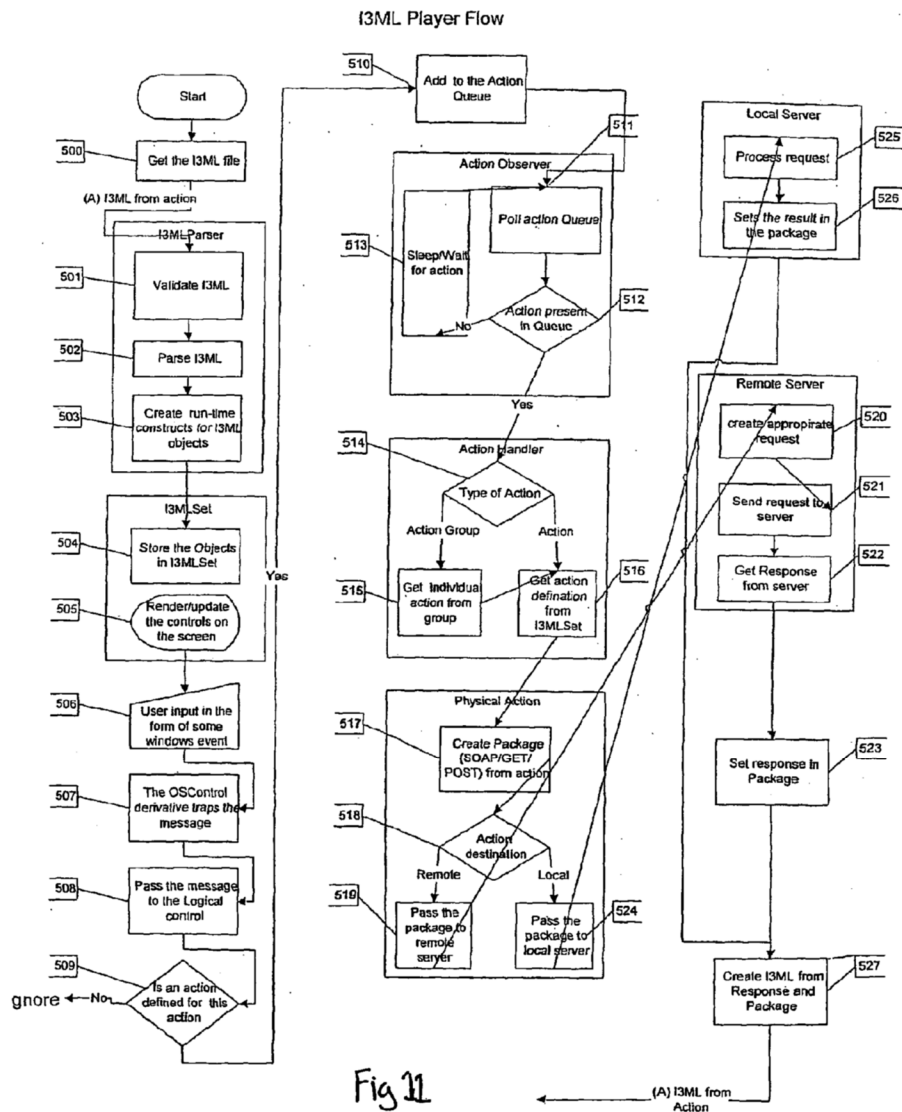
**D. The Asserted Claims of the '755, '287, and '044 Patents Are Not Obvious Over Stevens in Combination With Other References**

**1. Specific Issues with the Stevens and Rudolph Obviousness Ground**

1042. With regard to Stevens and Rudolph, Mr. Schmandt repeats many of the same general deficiencies that exist in his analysis of Huang and Rudolph. In my opinion, the POSA would not have been motivated to combine Stevens with Rudolph because the two target different audiences and use different technologies that run crosswise with each other.

1043. For example, similar to Huang, Stevens is targeted at businesses. Stevens’s focus is on business applications and on trying to provide a solution that would work within the





(SHOP0264111).

1107. Moreover, as discussed above, Mr. Schmandt fails to show other claimed features of the claimed UI Object, for example, that the information representative of the alleged UI Objects is stored in a database.

### 3. Huang and Stevens Obviousness Ground

1108. I have reviewed Mr. Schmandt's arguments that the asserted claims of the '755, '287, and '044 patents would have been obvious over the combination of Huang and Stevens. I disagree with Mr. Schmandt's opinions in this respect, and my opinion is that the asserted claims

of the '755, '287, and '044 patents would not have been obvious over the combination of Huang and Stevens. I note that Mr. Schmandt does not address all of Huang's deficiencies in his obviousness argument for this combination. I incorporate all of my previous discussions of both the Huang and Stevens reference here, and I address Mr. Schmandt's additional arguments below.

**a. General Comments**

1109. With regard to Huang and Stevens, Mr. Schmandt repeats many of the same general deficiencies that exist in his analysis of Huang and Rudolph, as well as Stevens and Rudolph. In my opinion, the POSA would not have been motivated to combine Huang with Stevens because the two use different technologies that run crosswise with each other.

1110. As discussed above, Huang and Stevens are directed to different methods and systems relating to providing different functionalities. Huang uses standard web technology that can be run in a standard browser in conjunction with a combination of a Web Service Player and a Web Service Factory Server, whereas Stevens uses a proprietary programming language that can only be programmed by its proprietary designer module and runs in its proprietary player module in conjunction with its proprietary I3ML server.

1111. More specifically, Huang relies on open, web-technology where the client portion runs in a browser and interfaces with Java Server Pages (JSP) or Java Server Faces (JSF) pages on a server. Steven on the other hand is closed and proprietary. As discussed above, although Stevens adopts the XML syntax for its programming language, called I3ML, I3ML is still a proprietary language that can only be executed by Steven's proprietary I3ML player module, which Mr. Schmandt acknowledges. (*See* Schmandt Rep., ¶ 1342 ("This is because browsers would not be able to understand I3ML, which was invented in Stevens, and for which Stevens

does not disclose any other code that can interpret the I3ML files to display the interface items.”)).

1112. Beyond the failure to articulate a specific motivation to combine or reasonable expectation of success, a key flaw with Mr. Schmandt’s Stevens and Huang obviousness ground is that it does not account for the claimed application and player, and their attendant features (such a device-dependence, device-independence, separation, etc.), required by the asserted claims of the ’755, ’287, and ’044 patents.

1113. Finally, Mr. Schmandt does not articulate any argument pertaining to reasonable expectation of success. Mr. Schmandt’s argument is fundamentally flawed because the proposed combinations would require one of the two systems to abandon its programming language and player module as well as the underlying architecture. To be clear, contrary to Mr. Schmandt’s characterization, Huang uses standard XML consistent with standard web technology and not a proprietary language with an XML syntax as Stevens does. (*See* Schmandt Rep., ¶ 1350).

1114. In fact, Stevens expressly discourages the use of browser compatible technology, such as Java or HTML. (*See, e.g.*, Stevens at [0007] (“However, JAVA applets, although designed for use over the internet, are frequently difficult to distribute and use owing to non-uniformity of JAVA implementations in web browsers.”); [0012] (“Though easy to develop and distribute, HTML does not provide a high level of functional richness.”)). Mr. Schmandt provides no explanation as to why or how this is possible, or how doing so would not undermine the core functionality of either Huang or Stevens.

- b. Element O –“build an application consisting of one or more web page views from at least a portion of said database utilizing at least one player, where said player utilizes information stored in said database to generate for the display of at least a portion of said one or more web pages,”**

1115. As a preliminary matter, and discussed above, Mr. Schmandt fails to establish that the player of Stevens is the claimed device dependent code. Moreover, for this claim limitation, Mr. Schmandt fails to explain how a POSA would modify Stevens's player module, which is a standalone, desktop application (e.g., "player.exe") that can only process the proprietary I3ML language, can be modified to work within Huang's browser-based player environment with its own combination of Web Service Player and Web Service Factory server.

1116. Mr. Schmandt summarily concludes that a POSA would be motivated to use the Stevens player because a player developed to run directly on the OS would have greater rights to run on the computer than browser-based applications and "could provide different functionality that may be desirable." (*See* Schmandt Rep., ¶ 1350). But Mr. Schmandt does not identify what different functionality would be sufficient to motivate a POSA to make attempt to do a whole-sale rewrite of a browser-based application as a standalone, desktop application. In essence, his analysis is driven by hindsight and using the claims as a roadmap.

1117. Therefore, Mr. Schmandt fails to show why a POSA would be motivated to make the asserted modification, what that modification would look like, and why a POSA would have a reasonable expectation of success.

**c. Element Q – "produce a Player, where said Player is a device dependent code;"**

1118. For this claim limitation, Mr. Schmandt repeats his argument verbatim from Element O immediately preceding. I, therefore, incorporate my discussion of Element O in response.

**d. Claim 17 of the '044 patent – "The method of claim 15103, where said web component is a text chat, a video chat, an image, a slideshow, a video, or an RSS feed."**