Data structure for SurfCast Grid

The data structure below is the representation of a SurfCast Grid in the first basic implementation. It is written in a pseudo-code which is largely similar to HTML. Certain features described are not part of standard HTML, but could be made available through the use of client browser plug-in modules. The syntax would have to be adjusted for such plug-in modules.

```
<HTML>
<HEAD>
<TITLE>Surfcast Grid Example</TITLE>
<META NAME="Ressource Manager" CONTENT="RESMGR.EXE">
<META NAME="Author" CONTENT="Surfcast, Inc.">
</HEAD>
...
<BODY BACKGROUND="surfback.gif">
<TABLE>
<TR>
<TD><A HREF="http://www.somewhere.com/sometarget.html"
       TARGET="new"
       ONMOUSEOVER=action1(arg1)
       ONMOUSEOUT=action2(arg2)
       REFRESH=timeout(200)
       SOURCE="http://www.surfcast.com/tilelibrary/tile2.til"
</TD>

TD><A HREF="http://www.somewhereelse.com/someothertarget.html"
       TARGET="new"
       ONMOUSEOVER=action3(arg3)
       ONMOUSEOUT=action4(arg4)
       REFRESH=timeout(500)
       SOURCE="http://www.camelot.castle/swords/excalibur.til"
</TD>

<TD><A HREF="local/document.htm"
       TARGET="new"
       ONMOUSEOVER=action1(arg3)
       ONMOUSEOUT=action2(arg4)
       REFRESH=timeout(0)
       SOURCE="local/documents/somedocument.til"
</TD>

</TR>

</TABLE>
</BODY>
</HTML>
```

Figure 1 - SurfCast Grid Document

Comments:

The main behaviour of the SurfCast Grid is given by the representation source (keyword "SOURCE") then the various action functions triggered by mouse events (keywords "ONMOUSEOVER" and "ONMOUSEOUT" or timer related events (as set by the "REFRESH" keyword). This will allow the grid contents to respond to user maneuverings and time. The "HREF" keyword decides what happens if the user actually selects the tile.

Further Data Structure Descriptions for SurfCast Tiles

A tile description in it's most basic form is already shown in the above example, but additional Tile behaviour could be added by further keywords:

```
<TD><A HREF="tile2_target.htm"
       TARGET="new"
       ONMOUSEOVER=action1(arg3)
       ONMOUSEOUT=action2(arg4)
       REFRESH=timeout(500)
       SOURCE=http://www.camelot.castle/swords/excalibur.til
       CLICKMAP={  0, 0,50,50, clickfunction1( clickargument1),
                  51, 0,50,50, clickfunction2( clickargument2),
                   0,51,50,10, clickfunction3( clickargument3) }
       TOOLBAR={"local/toolbars/radio.too" PLACEMENT="bottom" }
       ALARM="alarms/condition_rain=TRUE, condition_weekend=FALSE,
              alarmaction=blow_the_horn"



</TD>
```

Figure 2 - Further Tile Data Members

Comments:

The behaviour of a Tile can further be configured by adding function specific keywords. In this example, the Tile has a clickable map, i.e. separate areas of the tile produce separate results and actions when clicked. Also, this tile has a toolbar, which may appear only when the tile is in focus (has the mouse pointer over it), and provide special buttons or dials for adjusting the tile feature, such as sound or picture. Further, this tile has an alarm attached, so when triggered, this tile will change it's appearance, sound or functionality.

Client / Server conversation with multiple data streams

In order to describe this property of our invention, we assume a scenario with one client, who is speaking to one server where the client is requesting access to several different datastreams, and subsequently changing the properties of one of the datastreams.

The implementation of this functionality is in SurfCast Client Stage 2, and the corresponding SurfCast Server. The actual target data streams may reside anywhere. Only the properties of the data streams are controlled by the SurfCast elements, not the content of the datastreams themselves.

SURFCAST CLIENT		**SURFCAST SERVER (OR RESOURCE MGR.)**
LOGIN(USERPARMS)	→	
	←	ACKNOWLEDGE LOGIN(LOGINPARMS)
REGISTER(RESOURCE)	→	
	←	ACKNOWLEDGE REGISTRATION
REQUEST DATASTREAMS FOR STREAMS(STREAMPARMS)	→	
		VERIFY AND CALCULATE STREAMPARMS
	←	ACCEPT REQUEST (STREAMPARMS)
START DATASTREAM 1 START DATASTREAM 2 START DATASTREAM 3 (sent to datastream servers, not surfcast server)		

In the above example, the client logs in to a SurfCast server, and is verified. The client registers one or more resources, such as connection bandwidth, cost per transmission unit, durations, properties of local playback device and so on. The server can then verify these resources and compose the appropriate

The STREAMPARMS structure contains both overall client information and one or more stream request information blocks. Any number of streams can be requested in one request, the corresponding stream request information block sets/gives parameters for each stream.

Upon registration of RESOURCE the Server knows the client characteristics, and the Server is able to distribute bandwidth available to the client among multiple content servers.

So in this example, if the client has an incoming bandwidth of say 56Kbits/second, and he is requesting datastreams from three sources with equal priority, then the Server will respond in STREAMPARMS for each request, that a bandwidth of 56/3 = 18.7 Kbits/sec is available for each stream.

The actual end-to-end bandwidth obtained may be recalculated and readjusted on the fly.

In this example, the Server component may reside locally on the client machine (in which case the Server is known as the "Resource Manager", or it may be a remote SurfCast server.

One example of such architecture where a client informs the server of incoming bandwidth desired, is RealPlayer by RealNetworks, where this is exactly the way the client and server interacts. The parameters desired by the client is embedded in the connection request, so the server knows immediately how much to transmit to the client.

In a further implementation of the SurfCast software components, bandwidth calculations may be adjusted with empiric results in order to compensate for general of temporary bottlenecks outside the control of the SurfCast components.

Also, the STREAMPARMS may contain more than one bandwidth resource, say if the client has both an ISDN internet connection, another ISDN direct line available and is able to receive a satellite signal from the ASTRA satellites. In this

case, the calculations will cover both access, availability and cost considerations when determining the from a content server to a client.