White Paper
Scaling Interactive Television

Dan Lovy
August 2015

Abstract

Television audiences increasingly engage content on multiple screens and incorporate social media into their viewing experience, and "appointment television" has given way to cord-cutting and on-demand viewing. These seismic shifts are forcing content producers and advertisers to redefine storytelling and audience engagement. Cutting-edge television producers now strive to tell multi-season stories with direct tie-ins to games, apps, and social media. The lines are blurring between episodes, webisodes, games, ads, websites, and apps.

The media distribution world is shifting to a digital, one-to-one distribution model for content. This is creating opportunities to exploit the ability to create unique and personal content and add a two way communication between the viewer and content.

The problem is, there is no standard, defined way to create any of it; it's all being done one-off, poorly, or not at all. It is urgent that the industry define ways in which content producers can create these experiences and build a compatible set of supporting tools, technology, and infrastructure.

SIMC (Social and Interactive Media Consortium) is an industry group seeking to create a protocol that solves this problem.

This white paper focuses only on scalability. The issues is not scaling the volume but rather the challenges of scaling content across infrastructures, devices and work flows that would enable economic publishing at large scale.

Scalability Problem

When a video publishers creates content, they are not concerned that whether broadcast and viewing hardware can display NTSC video. Likewise, all the tools for recording, editing, storage and distribution of video content adhere to the same sets of standards. A publisher is not concerned with interoperability across anything.

The same is not true when interactivity, especially interactivity across devices is added to the viewer's experience. These systems have tended to be highly proprietary and focused on specific hardware and specific networks. In order to scale, content producers need tools that can create interactive video experiences that are completely independent of the viewing environment. Content should run on ROKU boxes running NetFlix to set top boxes running Comcast servers. Input can come from phones, tablets or remote controls.

Any standard protocol solution must scale across platforms.

The SIMC Solution

Protocols often involve the invention of languages. These languages are designed to better represent information in target domains. A domain language typically creates and manages data that codes for the desired behavior. The drives all the tools needed to create and publish content.

PostScript for example, is such a protocol. It defines a language for resolution independent page description, especially for type fonts. This solution works well as the mathematics of font representation do not change over time.

The same cannot be said for interactive video content. One representation might work well for one style of content, but be unworkable for another. This industry is in its infancy and it is important to provide an expanding palette of capabilities. It is also key that publishers be able to invent different categories of content complete with appropriate tools and work flows.

Part of the solution SIMC solution borrows a page from micro code found in modern CPUs. Micro code is a low level machine instruction language. This language is used to build the CPU's instruction set. The SIMC solution uses Java Script as a low level micro code. Now, different domain languages can be created depending on content requirement but still run across multiple platforms.

Example

To illustrate how this works, let's start with an example.

Imagine HBO has decided to create a multi-user, group based, interactive, multi-threaded story based on the Game Of Thrones universe.

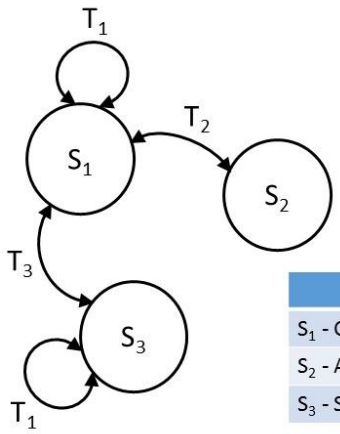
Viewers form groups of 6. Each viewer is one particular character. Groups can be randomly assigned or allowed to select themselves. The story(s) are then told from each characters point of view.

At different points in each character's story line the viewer is asked to make a simple character choice, form an alliance with another character or betray someone. After that decision, everyone's story line could be effected. These choices are offered either directly on screen or through another device.

Yes, this would require a great increase in footage, but it could encourage a great deal of engagement and multi viewings as viewers play out different scenarios and try different characters. Ten linear episodes would have to translate into ten times the footage but result in twenty times the individual viewership. There would be a viewing data feedback loop to manage this.

The creator has a complex universe to manage but can create a different style of emotional engagement. Selection of the choice points and intertwining story lines is a new and different story telling language.

It is easy to imagine a domain language whose content representation is a state transition table.



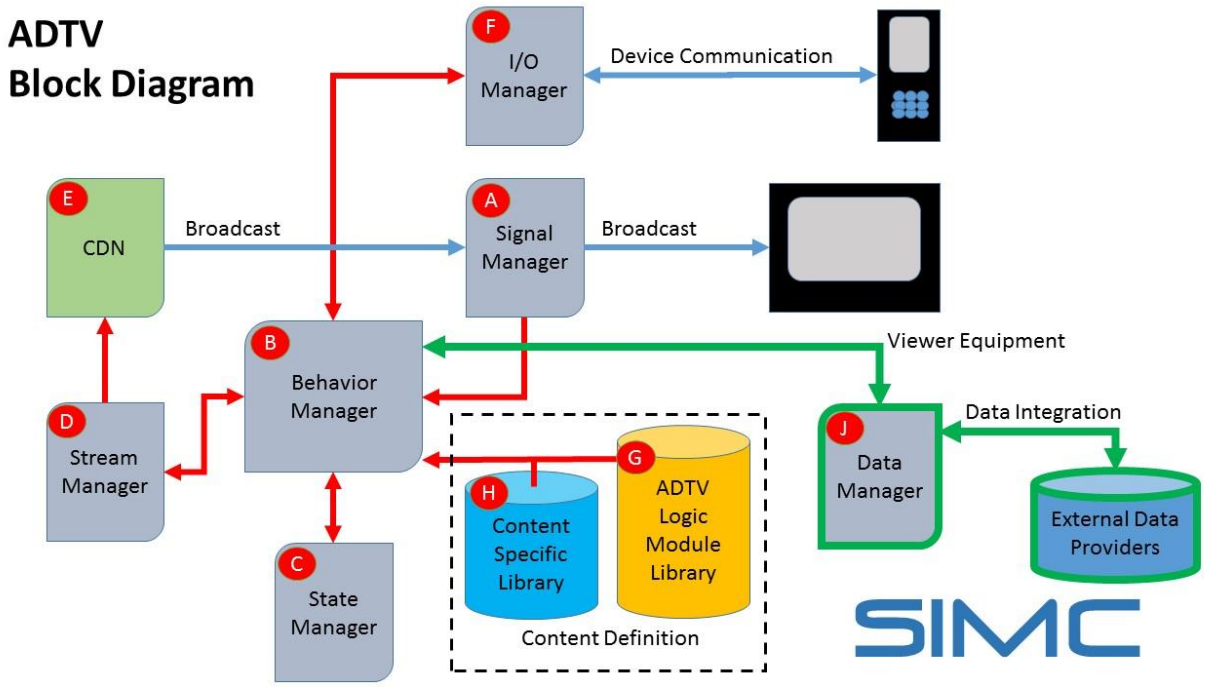
	T_1 – Sleep	T_2 - Struck	T_3 – Yelled
S_1 - Calm	S_1	S_2	S_3
S_2 - Angry	UNDEF	S_1	UNDEF
S_3 - Sad	S_3	UNDEF	S_1

State Transition Table

There would be tools for creating, editing and testing state transition tables. Interpreting these tables would generate the viewers experience and distributing these tables is the how, in part, this content would be published.

ADTV

SIMC's protocol is called ADTV (Advanced Digital TV). Here is a high level block diagram of how the overall protocol operates. This is from the ADTV functional specification.



I will focus on three components; (B) Behavior Manager, (G) ADTV Logic Module Library and (H) Content Specific Library.

To use the micro code metaphor, the Behavior Manager functions as the hardware CPU. It is communicating with other components of the system. Its behavior is controlled by code in the ADTV Logic Module Library. These Logic Modules act much as micro code in a CPU. They would form the rules that govern the interpretation of a data language. In the Game Of Thrones example, these would be the interpretation rules of the state transition tables.

The actual tables, the specific instructions laid out by a content creator are stored in the Content Specific Library. These instructions are fetched during the play back of the content with each instruction being interpreted by code in the Logic Module. As mentioned before, this code is highly portable Java Script.

Now it's off the races. Increasingly sophisticated tools and authoring environments can be built to support this style of content. The complexity of the stories is not bounded by limitations of the language but by the quality of the story teller's imagination.

Two problems still remain. How do you insure interoperability on different platforms and what happens when there are content models that don't represent well as state transition table?

Interoperability

Content is represented by data. How do you guarantee that this data will be interpreted consistently on different platforms? The interpreting code is written in a portable format, Java Script and accessed from a central library of code.

There is information in the content itself that specifies which Logic Module code to use. All platforms reference the same libraries. This makes sure that the rules by which content behavior data is interpreted is independent of where that content is running.

The designers of the data language are responsible for making their code available in this library. Fixes can be published to this library. New features and extensions can be created centrally without having to work directly with each platform that has adopted this standard.

Content with a similar style can now be created in authoring tool environments. The output is always data and the interpretation rules are known.

Different Content Styles

What if an interactive design does not fit the Game Of Thrones style of state machine management? Key to the success of this protocol will be the ability to create new categories of experiences and have a publishing model that works at scale.

New code can be added to Logic Modules. This code could provide additions to existing functionality or introduce a completely different model. Tools can be modified to take advantage of new capabilities are complete new ones produced that give different creative dimensions for producers to explore.

Interactive television is a new medium. There is a protocol standardization challenge but more importantly there is a need to provide a consistent publishing platform and create opportunities for invention. SIMC is bringing together contributors from a range of companies, each helping fill in the story.

For more information, visit www.simc.tv

Or email

Dan Lovy at dan@simc.tv