New Methodologies in Real-Time Data Aggregation and Management for Broadcast Presentation and Distribution

Alain Savoie, Vernon Freedlander, Georg Hentsch
Bannister Lake
Cambridge, Ontario

Abstract - *Live data and its management have become an integral part of the broadcast workflow. Whether it is used in a fast-moving dynamic presentation such as elections, sports or finance, or used to populate and control on-air branding, effective live data aggregation and management drives efficiencies and creates new monetization possibilities. As broadcasters seek out new revenue streams, live data can play a more significant role on-air, online and in developing new markets such as digital signage. This paper will examine how broadcast engineers and IT technologists can best contain, aggregate, reformat and repurpose live data from multiple, diverse sources to generate more efficient workflows and ignite revenue opportunities. The paper will also discuss how live data integrates with other associated equipment in the broadcast plant to create a more desirable workflow. Examples of recent projects in broadcast, large scale eSports events, and in digital signage will be used to illustrate.*

THE POWER OF METHODICAL DATA CENTRALIZATION AND MANAGEMENT

Broadcasters have always understood the value of data. Ratings, demographic insights, and performance analytics have driven our industry. On-air data, be it in news tickers, populating graphics for branding, or driving AR and VR solutions has also played a significant role in day to day production. The concept of "big data" has always been a part of the broadcasting landscape and continues to thrive today as new services launch online and machine learning grows in influence.

Data is increasingly playing a growing editorial role. From social media integration to web widgets to interactive audience polling to second screen applications, data is becoming an important part of the "story." Live data drives ever-sophisticated augmented reality elements primarily in news, weather and sports. The storytelling power of real-time data drives audience engagement and plays a vital role in keeping viewers informed and up to date. Real-time data gives news, sports, weather, and financial broadcasters the sense of urgency and authority that makes them competitive and relevant in the market.

Aggregating and managing data for on-air usage has had its challenges, both operationally and financially. The concept of a data frontend that populates graphic templates via a graphics engine is not new, but the traditional approach has not kept up with broadcasters' and visual communicators' demands for data content. Typically, broadcasters are required to purchase "data modules" depending on the specific data feed that needs to be integrated. If a broadcaster requires tickers that behave in a specific manner, extensive custom development is required. This is expensive, time-consuming and requires highly specialized personnel. Data has also tended to be siloed within the broadcast plant with the on-air side, online side, sales, traffic, and programming not coordinating resources.

Bannister Lake has put a new model forward that focuses on centralizing data aggregation and management and making data management accessible through any web browser. The approach rethinks the modular approach and reduces or eliminates the need for custom development by establishing a set of tools within the solution to build data and graphics solutions.

The methodology underlying Bannister Lake's Chameleon product is two-fold; to serve as a single, centralized solution for data aggregation that can be utilized throughout the broadcast and graphic display ecosystem and secondly to reorganize various incoming data feeds into an elegant, comprehensive database structure that makes the handling of data efficient and optimized.

This goes beyond broadcast and makes data content readily available to a variety of solutions including web widgets, AR/VR systems, streaming, digital signage, and online. A SQL-based query module allows for an unlimited number of data queries resulting in specific data subsets based on conditions. This approach dramatically reduces custom development time and provides users with more on-screen functionality. Its RESTful API provides the ability to reformat data and output it to a wide variety of endpoints creating new business and editorial possibilities. Instead of forcing a broadcaster to obtain several data modules, the product is equipped to read an extensive variety of data sources. A "custom data module" opens the possibility to access and read additional data sources including Google Sheets. The integration of Google Sheets into the data workflow creates unprecedented collaboration opportunities among production staff. In a breaking or developing news situation, this provides the user with the ability to spontaneously use a variety of data sources to populate graphic templates and tell a compelling and multifaceted story and quickly get it to air. In a non-news environment, a custom data module can be used by staff who have limited technical knowledge to distribute and display topical information for special events and digital signage applications. In concert with this approach is the concept of public access to the data aggregation solution. A centralized system used in a broadcast or signage environment should allow "whitelisted" members of the public to add content. This may take the form of school closings, or community alerts or opinion polling.

Beyond daily news coverage, a centralized data aggregation solution provides a wide variety of benefits. Broadcasters can effectively amortize and manage their data products across multiple platforms. A branded data feed originating from the station or network can be distributed to customized web widgets, a digital signage network, a second screen application or a streaming service.

The same centralized solution can integrate advertising and take advantage of traffic systems to trigger events. It is crucial that any on-air data solution be able to communicate with traffic systems and coordinate with sales and programming. As such, the solution must provide the ability to include program data and to generate as-run logs to satisfy sales.

This concept of a fully integrated data solution that incorporates and manages not only on-air tickers, branding graphics, and full-frame graphics, but is also open enough to coordinate with other broadcast systems is essential for machine learning possibilities. By coordinating the widest variety of data sources available, both on-air and administrative, broadcasters can develop rule-based machine learning solutions. A centralized data aggregation and management solution is an integral step towards this goal. A sophisticated AI solution would trigger editorial, brand or sponsorship related content based on events, thresholds or data analytics leveraging automation.

Engineering Data Centralization and a Better Database Structure

Over years of developing data and graphics solutions for elections, sports, news and weather, it became apparent that there is no standardization of data. For example, each data source for sports scores has a different format and often can't be mapped easily into the ultimate schema. Most data may have some fields in common such as teams, scores and game status but that is a tiny subset of a typical data requirement.

Each sport has unique data associated with a game score. For ice hockey, a requirement may be shots on goals or goalie save percentage while baseball has batter, pitcher, count, outs, pitch speed, and so much more. For this reason, having a fixed database table to contain all data is not practical.

This may lead to building a custom database schema for each data requirement and in fact, that would contain data most efficiently. However, it's never enough to build a database or even populate a database programmatically. One would require a user interface to CRUD (create, read, update and delete). Also required is a way to filter, organize and format the data. These requirements would have to be multiuser and preferably web-based since the industry has moved far beyond a windows-only world. The target platform would most likely include PCs, Macs, tablets, laptops, Chromebooks and phones.

It is practical to assume budgets, both financial and time won't allow custom development for each project. This leads to a specific requirement for containing all data in a single database schema. The approach methodology is to have core data types:

Figure. 1: Core Editorial Data Types

And attach data to the standard data types using key/value pairs.

The methodology is to take a standard relational database design and attach key/value design to provide the flexibility required. Databases like MySQL have this built in with the json data type and could be considered if the goal is to be tightly coupled to a single database. But creating the key/value support is easily done by adding companion tables which provide all that is required for key/value pairs.

For example, for a standard data type such as editorial stories from sources such as the AP or Reuters, the following fields may appear:

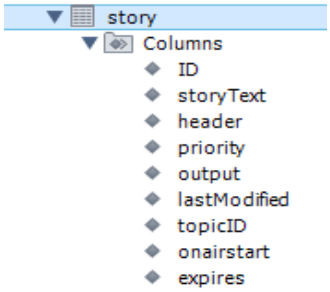

Figure. 2: Standard Data Types For Editorial Stories

This approach provides support for the story, headline (header) and several standard fields like expiry, last modified, and a container (topic/topicID). But what if there is a need to attach additional data such as author, clickable URL or alternate language translations? One possibility is to continue to add new fields to the table but that's not practical because the schema should not change whenever new requirements are being added.

Instead, the approach is to add a new table as a companion to the core data type with a design like this:

Table: story_tag

Columns:

ID	int(10) UN AI PK
storyID	int(10) UN
tagID	int(10) UN
key	varchar(32)
value	text
type	varchar(16)
mediaID	int(10) UN
datafeedID	bigint(20) UN
datafeedChar	varchar(255)
lastModified	timestamp

Figure. 3: Companion Table

Here it references back to the core data type but adds key/value pairs using fields key and value. This by default provides all the flexibility needed to mix relational and key/value data under one umbrella.

This provides great flexibility to represent any type of data. In the case of weather alerts, developers can take weather from the National Weather Service and map it into a story. The headline is in bold while the story copy is below it. All the key/value pairs are displayed in the CRUD UI below the copy. The keys are event, status, message type, etc.

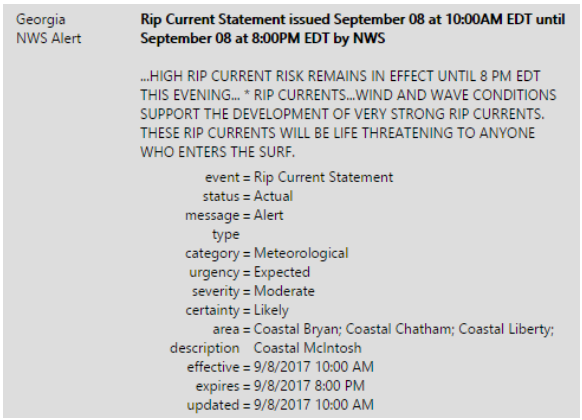

FIGURE. 4: HEADLINE WITH CRUD UI INCLUDING KEYS

But what if the data being contained doesn't map into one of the core data types? What if developers had a mostly empty data type which becomes a carrier for the key/value pairs? That is exactly the requirement methodology underlying the "custom data type." It's a catchall for containing anything. A practical example is the requirement for custom data to contain medal leaders from the Winter Olympics:

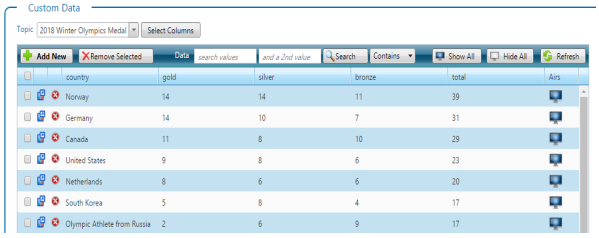

FIGURE. 5: CUSTOM DATA CATCHALL

The data is contained in the key/value table which points to the main relational data record:

ID	customID	tagID	key	value	type
1	1	NULL	rank	1	number
2	1	NULL	country	Norway	text
3	1	NULL	gold	14	number
4	1	NULL	silver	14	number
5	1	NULL	bronze	11	number
6	1	NULL	total	39	number
7	1	NULL	rankth	1st	text
8	2	NULL	rank	5	number
9	2	NULL	country	Netherlands	text
10	2	NULL	gold	8	number
11	2	NULL	silver	6	number
12	2	NULL	bronze	6	number
13	2	NULL	total	20	number
14	2	NULL	rankth	5th	text

FIGURE. 6: DATA CONTAINED IN KEY/VALUE TABLE

Note that the solution does include some type information for the key/value pair. This is generally useful for graphics platforms which might have logic capabilities to format data. But it's not essential.

At this point, the solution is in a good position to contain almost any data that might be required. However, that's clearly not enough. Playlists can be used to organize data in a static fashion. A playlist is a manual way to organize records and is quite useful. For example, it may be used to pick certain stocks from a large inventory of quotes. It is quite useful but what if what is needed is a more dynamic mechanism? For example, what if a requirement was to show all NBA and NHL games that are in-progress? Or a requirement is to show the top 10 volume leaders from the Dow Jones components? What is needed is a way to create dynamic playlists.

This can be done by using queries. Given the data available, if the query mechanism is flexible, the data can be formatted as is required; not only the records, but ordering, formatting and limits. A good example of this happens in basketball (and all sports) where there is a requirement to contain every player and his/her stats in the NBA, but also a need to recall the top 10 blocks/game leaders in the Western Conference. Or perhaps production wants the top 10 steals/game leaders from players born in Spain or Lithuania. If the data is available, it can be queried.

The use of a mixed relational and key/value design, certainly, doesn't simplify the process, but with some basic SQL knowledge, users are now in a position to create dynamic playlists with very few limitations. For example, a typical query that would recall the top 10 heaviest players in the NHL would appear this way:

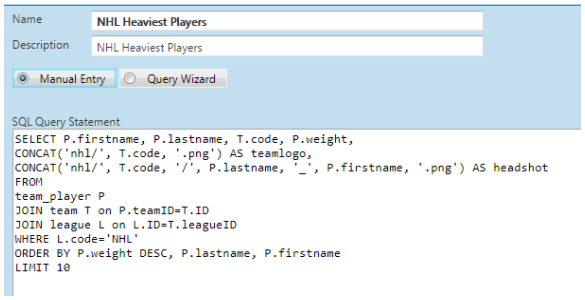

FIGURE. 7: TYPICAL DATA QUERY

Which gets formatted as follows including custom formatting for team logos and headshots.

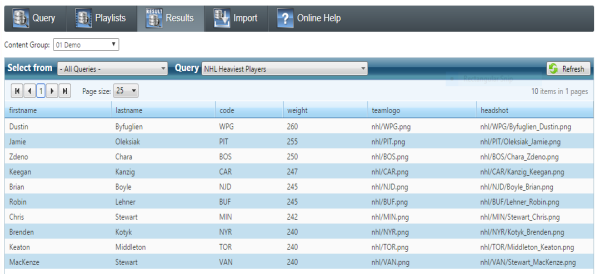

FIGURE. 8: QUERY FORMATTED

At this point, the solution can now contain most any data and have flexibility in formatting the data. However, there is a basic requirement to use this data across multiple endpoints, whether it be in a ticker, a CG, digital signage, website or published as a data feed. What is needed is a way of mapping any of the available data whether it's core data, static or dynamic playlist to common data formats. This might be json, xml, csv or whatever makes sense for the data.

The common way this is done is with a RESTful API. By using HTTP requests to GET data with the use of a web service, it is possible to pull any of the containers out of the database in standard formats. For example, a RESTful API would define the Olympics medal count example in json format as following:


```
{
 "generated": "2019-01-08 23:19:29Z",
 "customList": [
  {
   "id": 1,
   "topicBroadcastName": "Medal Count",
   "topicName": "2018 Winter Olympics Medal Count",
   "modifiedDate": "2018-02-11T12:46:46-05:00",
   "rank": "1",
   "country": "Norway",
   "gold": "14",
   "silver": "14",
   "bronze": "11",
   "total": "39",
   "rankth": "1st"
  },
  {
   "id": 3,
   "topicBroadcastName": "Medal Count",
   "topicName": "2018 Winter Olympics Medal Count",
   "modifiedDate": "2018-02-11T12:46:50-05:00",
   "rank": "2",
   "country": "Germany",
   "gold": "14",
   "silver": "10",
   "bronze": "7",
   "total": "31",
   "rankth": "2nd"
  },...
```


FIGURE. 9: RESTFUL API DEFINITION IN JSON FORMAT

The data can go beyond standard broadcast targets like tickers and branding. The data can be used in OTT or standard websites by using the data via the RESTful API through the use of web widgets. For example, the data can be made readily available to publish a data visualization to a broadcaster's website.

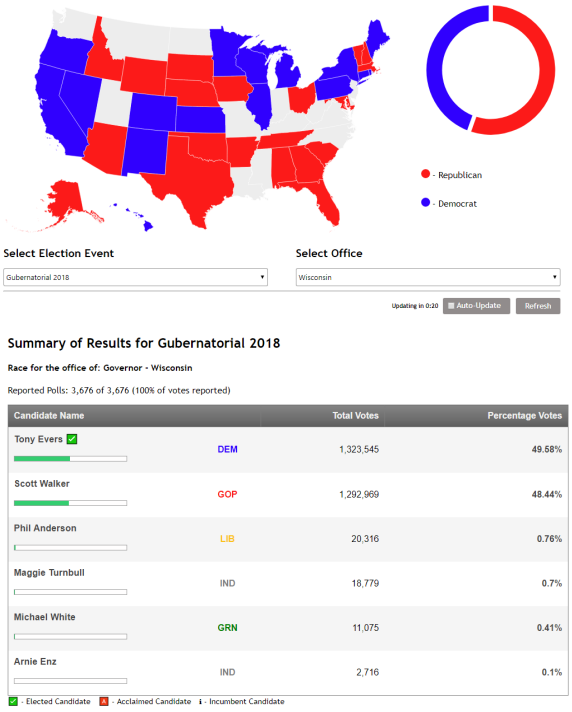

FIGURE. 10: WEB WIDGET DRIVEN BY REAL-TIME DATA

Typically, a RESTful API is a useful mechanism for a loose coupling to the target. Whether that target is a website, a ticker, a CG or digital signage, it becomes an excellent way to share the data. However, what if that data could also include tighter coupling? Tighter integration would include scheduling and creation of rundowns for use in tickers and branding. This is another way to utilize data and make it useful and easily formatted for a specific graphics requirement. For example, if developing a ticker with zones, not only would a requirement be datetime scheduling of the graphics/data, but also to define a rundown using any of the containers that are available in the

database including standard containers (story topics, sports leagues, etc.), static playlists, and dynamic playlists.
Tight coupling does require some graphics programming, but it allows for an easy way for users to define their graphics based on data, especially if an easy to use web interface is available to users. For example, this is an example for specifying the data that would be available to a ticker zone:

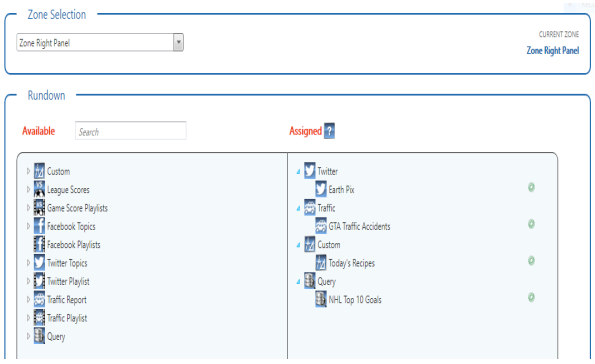

FIGURE. 11: DATA AVAILABLE IN A SPECIFIC TICKER ZONE

On the left, there is the available data and it's just a matter of dragging the data into the right column to create an order to the rundown. By having the data available to all devices/targets, but also with the potential for tight coupling to graphics, the platform is flexible enough for fulfilling most data/graphics/website requirements.

Bannister Lake is expanding upon this methodology to take this much further. Ultimately, data from multiple instances of the database should move easily between each instance. This movement of data can be from cloud to cloud, cloud to local, or local to cloud. Each of these databases have a precisely identical schema, so all that's needed is a mechanism to move data between instances. A natural evolution of the methodology is to try to create an engine for publishing data, while simultaneously having the data flow between multiple instances.

DATA AGGREGATION AND MANAGEMENT METHODOLOGY IN PRACTICE

Bannister Lake's centralized data methodology provides a broadcaster or event producer with a variety of practical data sharing benefits that can be used throughout the production and visual display ecosystem.

For station groups looking to create efficiencies around their on-air data distribution among various stations, the centralized system allows for a single on-location server or the use of the cloud to aggregate data from across all stations and all incoming data sources. Once the data is located on this single database instance, it is available to share among various content groups. The UI associated with the software allows for the formation of content groups that are specifically defined by region, station, department, or groups of individuals. Content within the content group can be organized into regional/topic related content, or specific content playlists.

Each broadcast outlet can then schedule their own localized rundown to showcase what content will be displayed and how it will behave on-air at any given time of the day. From a single database instance, this approach to configuration allows for an unlimited number of tickers generated across various platforms. As such a station group or network can manage tickers throughout their affiliated stations but still provide enough flexibility to allow a local station to manage and present its own distinct content.

The exact same approach provides similar controls over on-air branding. As a centralized solution networked to station/network traffic systems, it is able to not only hold data related content but can store specific data related to programing information. Because the system can read and import the station groups' or network's traffic schedule, other departments such as marketing can modify or add any supplementary information such as social media, or casting information, or even program associated art-work.

This content and its display can be controlled through the solution providing a centralized solution for streaming channels, apps and enhanced promotions. The solution is engineered to constantly monitor the program schedule at any time during the day to ensure that when automation triggers an up next promotional banner, the correct information along with the associated artwork will appear as a graphic on-air. However, because the data imported includes the entire station group or network, at any given time automation can trigger a graphic template that promotes show data from a companion channel. This feature is only possible when data content is truly centralized and a solution is provided for simplified and strategic cross branding.

A centralized approach to data extends itself to the production process. When producing high-profile, data-rich events such as elections, Bannister Lake has developed a new approach to organizing data that leverages centralization and fosters a more efficient way for producers to track and display results. Election results are aggregated from multiple sources and not just from one provider.

The underlying methodology is to then take these diverse datasets and organize them in databases that allow for exceptionally fast recall. The results are organized hierarchically by distinct event, for example the US Midterm Elections, followed by distinct races, for example US Senate seats. All the necessary editorial data required for an election show, number of votes cast, comparative vote percentages, candidate biographical information and other significant content is similarly organized within the solution's database. This methodology of organizing content then allows production staff, organized by editorial or functional groups, to create specific playlists of races they want to monitor throughout election night.

Because data is centralized and uniquely organized, playlists can be conditional and filtered more effectively, providing producers with additional editorial insights and deeper political analytics. The same playlists can be sent to the production's graphic system for playout or rendered out via the solution's HTML5 renderer. The system's API can also be used to reformat election data for integration into other production and display endpoints.

In non-broadcast situations such as major event digital signage where complex and fast-moving data needs to be aggregated and managed and ultimately displayed, Bannister Lake's data methodology simplifies the process while adding a number of production benefits. A typical example would be a major sports event where a wide variety and a high number of diverse datasets coming from many sources are required for analysis and display. This may include player related statistics, schedules, venue details, event generated news, status updates, weather, and player or team performance statistics, such as wins, losses, speed of pitch or serve, number of tackles, etc.

Similar to election data, the methodology is to take incoming data and reorganize it into a database that facilitates editorial recall, filtering, and efficient distribution to endpoints. It is by aggregating and then organizing incoming diverse data into a specifically defined hierarchical database optimized for recall, that allows for quick filtering, analysis and as required generate a unique, single XML URL for distribution. Similarly, playlisted data or individual datasets can be associated with a "match ID" allowing operators to quickly call up data and associated graphic content.

The same principles apply to multiplayer or multiteam events such as eSports. A unique feature of eSports is the high quantity of matches played at the same time involving hundreds of players. With all event related data stored centrally and utilizing the solution's unique organizing of data recall, filtering, and analysis of data content, its distribution is simplified.

CONCLUSION

Broadcasters and visual communicators have enormous opportunities utilizing data, both real-time and static, to create better journalism, communicate more effectively with audiences, and power automation and AI capabilities. However, if data is siloed, both technically and editorially, the cost of developing solutions, including data-rich products, will prohibit it from being incorporated fully into workflows and into the production/distribution ecosystem in general. It is only by centralizing data, and making it readily available throughout the broadcast plant, and by developing a process of structural standardization, that the full potential of data can be realized.

Centralization, with the added benefits of a RESTful API allows data to flow unencumbered between a wide variety of devices and endpoints. Reorganizing data leverages query functionality making the data not only easily searchable via SQL but allowing subsets of data to be identified and collected, driving new revenues and product potential.

Data has become the lifeblood of the digital economy but it is only through its proper and consistent organization and management that the communications industry can take full advantage of the possibilities it offers.