A Semantically Powered Digital Audio Workstation in the Browser

Nicholas Jillings and Ryan Stables

1 DMT Lab, Birmingham City University, Birmingham, UK

Correspondence should be addressed to Nicholas Jillings (nicholas.jillings@bcu.ac.uk)

ABSTRACT

In this study, we present an online music production tool that facilitates the capture of time-series audio and session data, including action history. This allows us to analyse sessions and infer production decisions based on actions made to the user interface. We conduct an experiment in which mix engineers were asked to use the system to perform a balance mix, then we provide observations made using the system. We show that participants often exhibit commonalities in mixing styles when applying gain and panning to specific instruments in a mix, and demonstrate common temporal characteristics relating to the magnitude of parameter adjustments.

1 Introduction

Producing music includes a number of complex tasks, and typically involves achieving an artistic vision using engineering skills. Music production has undergone a digital revolution, in which a large number of analogue techniques have computational equivalents of emulations. Despite this, user interfaces have changed very little, and production is still heavily reliant upon the expertise of the mixing engineer, where they are required to translate an artists’ inexperienced description of their vision into a set of parameters and controls.

1.1 Intelligent Music Production Tools

Intelligent production tools aim to help bridge the gap between the engineer and the artist by automating and facilitating a number of processes [1]. Early examples of work in the field focus on automatic channel mixing to control conference microphones [2]. More recently these have progressed into automatic engines, modifying multiple parameters such as faders [3], panning [4] and EQ [5]. These techniques are based on raw information from the signal to derive instantaneous results. Semantic music production environments [6] apply similar processes, but attempt to reason parameters by learning external factors, such as the genre and tempo of the song, the instrument on a track, or an engineer’s description of desired the musical timbre.

Intelligent mixing systems often require a knowledge base of expert user inputs, which is then incorporated as a set of predefined rules. This often works for a specific subset of mixes [1], but can create unwanted bias in the resulting mix. Intelligent systems can also be personalized to an individual user’s preferences (e.g. [7]), as opposed to a generalised system of rules. This allows the tool to learn common practices, thus making the workflow more intuitive.
1.2 Music Production on the Web

As semantic web technologies and linked data become more prominent, tools for online audio processing are able to perform more intelligent functionality due to the retrieval of large amounts of music production data. To facilitate this, frameworks for understanding the various mixing processes exist as ontologies [8, 9]. Where graph-like data structures can be used in knowledge-based systems to infer relationships between components in the production process.

In this paper, we present a Digital Audio Workstation (DAW)\(^1\), built in the browser to specifically gather data on mixing practices for intelligent functionality. The following sections will outline the DAW and database, followed by details of a mixing experiment, conducted to demonstrate the capabilities of a music production system on the web.

2 System Overview

A semantically powered DAW enables knowledge-based systems to be deployed directly into the audio production process. By restructuring the standard host architecture, we are able to develop a fully functioning environment, specifically designed for the collection and utilisation of semantics data. This builds on systems that currently address semantic processing at the audio processing [6, 7] or mixing stage [3, 4].

2.1 User Interface

The timeline view (presented in Fig. 1) and mixer view are both based empirically on traditional DAW interfaces, providing users with a familiar working environment. This allows us to gather data relating to traditional DAW usage, rather than exploring novel interfaces such as [10] as standard.

The user-interface is based on the Model-View-Presentation software architecture, where the model is an empirically developed audio engine and the view can be a timeline or mixer window. The audio engine has a predefined set of objects which the view must translate into graphical elements for the presenter. The user interacts with the on-screen elements, which are translated into control signals for the audio engine (model). We use web tools such as AngularJS\(^2\) and Bootstrap\(^3\). This way, the code for the front-end becomes manageable and easy to expand on in future versions.

2.2 Database Design

In our database, all aspects of the audio production environment are classed as objects with a finite set of variables (based on [11]), and can be captured with their predefined relationships to other objects. Ontologies exist to define known objects and artefacts of the mixing process. [8] and [9] are aimed at defining the

---

\(^1\)http://www.semanticaudio.co.uk/webdaw

\(^2\)https://angularjs.org/

\(^3\)http://getbootstrap.com/
mixing process for effects processors and physical studio lifetimes. Ontologies are less useful for storing data due to their large amount of often redundant fields, so instead we use a traditional relational database for the session data, with relationships captured in tabular form.

Figure 2 illustrates a subset of the database with key relationships between a number of selected objects. Each object can be tracked back to the root session node and can be represented in a relational format. For instance, a track has an audio region, therefore it can also be represented as being an “on track” object, loosely based around the concept of a triple store. The database also captures the history of every object allowing for detailed analysis of the mix process to be discovered.

2.3 Audio Engine

The Web Audio API [12], introduced as part of HTML5, brings real-time audio processing natively to the browser. All major mobile and desktop browsers support the API4. The API operates by linking audio nodes together, creating a processing graph which the browser renders.

The DAW uses the API for audio processing and management of events. It defines the audio regions, tracks, busses and other aspects of the mixing environment, highlighted in Fig. 3. This audio engine links to the database to communicate any data changes, such as changing a track’s volume or a region’s start time, since the objects defined by the audio engine are the same as those defined in the database (Figures 2 and 3).

Figure 4 shows the track and bus object structure. Each bus has a traditional, linear view from the input gain on the left, down through the effects chain to the outputs. The mute and solo control nodes are omitted for clarity. The audio engine also incorporates JS-Xtract [13] nodes to enable real-time feature extraction for visuals and other data collection. For instance, the volume meters use the RMS amplitude feature on the output of the channel.

4At time of writing, Safari is using an early version of the API. http://caniuse.com/#feat=audio-api

2.4 Plugin Architecture

To support audio effects and synthesis modules, we use the JSAP web-audio plugin format [14], developed specifically for web-audio-API-enabled platforms. It defines both the host interface and the plugin interface. The interface allows plugins to interact with other web audio nodes as if they were a single node, rather than a collection, and exposes parameters with boundaries and various data types.

The format allows the host to define a PluginFactory, where the plugin constructors are stored to build individual instances on request. The factory also spawns SubFactory objects which hold a plugin chain, similar to the linear effects chain found in traditional DAWs. The PluginFactory also facilitates inter-plugin communication for the sharing of features and session and track information. This allows inline adaptive and cross-adaptive processing, without a requirement for additional routing or side-chaining.

3 Session Structure Analysis

To utilise the capability of the system, we conduct a mixing experiment in which we capture a session data and provide a novel overview of mixing practices under various conditions. This is then used to create default
Fig. 4: Structure of the track object and bus object in the audio engine.

session structures, providing mix engineers with a more intuitive starting position. In our system, it is possible to capture every decision an engineer takes when building the mix. This provides us with time-series session data, allowing us to evaluate ways in which engineers adapt during the mixing process. This will contribute to the relatively few datasets that currently only capture static session data [15].

To perform a mix, an engineer will start with a set of recorded unprocessed stems, and progress through a number of corrective and creative sages to achieve a final fader and pan balance. This is often referred to as a balance mix [16], which is an unprocessed mix using just the faders, panning and routing to define the layout of the song. The mixing process can be conceptually perceived as an n-dimensional environment where every possible permutation of a mix can be explored [11]. Most DAWs start the mix at the origin, with everything set to unity (pan is set to centre, the fader is at unity gain and all tracks feed the master bus). Interestingly, this starting position informs the actions of the user, as varying the initial gain structures often produces variation in the final structure [17].

3.1 Previous Work

The aim of a balance mix is to set the fader positions, to adjust the pan positions for each channel and to route channels to appropriate groups and busses. This process has been shown to have commonalities across mix engineers for a given genre of music, such as the prominence of certain instrument types in the mix. For example, [17] show that vocals are consistently louder than other sources in a mix, with an average relative level of -3 LU. Similarly, spatial processing in the mix tends to be dependent on the genre of the music, where within-genre trends are often observed between tracks [18].

In addition, the timbre of sources often informs the mixing decisions. For example, channels which are predominantly low frequency are panned centrally, with the bass guitar almost always centrally panned, whereas higher frequency sounds tend to utilise spatial positioning. Furthermore, the starting position of the faders in the mix has an impact on the final mix decisions [17]. Several engineers mixing the same song with two different starting positions tend towards a local minima in the conceptual mix space. Routing and grouping of channels is also performed in the balance mix stage [16]. Tracks are often grouped together based on their instrument type [19]. It is also common for groups to be further grouped (sub-grouping) to create sub-mixes for either processing or control. Songs with more tracks tend to have more groups. Also mixes with more groups tended to result in perceptually higher quality mixes.

3.2 Mixing Experiments

To conduct our mixing experiments, a group of subjects with experience in music production5 were presented with the empirically developed online DAW6 and asked to produce a balance mix in their home studios. They were first shown a loading page, with a list of 5 songs to mix, taken from the Open Multitrack Testbed [15]. Each song was a 30 second excerpt of the chorus of each track to save both loading and operational time for the end users. To enforce consistency in the data, the next song could only be mixed when the preceding song was completed.

To constrain the experiments, the DAW was limited to only allow actions which are related to the creation of the balance mix. This includes changing the volume, panning, creating busses and configuring groups

5Predominantly undergraduate students studying sound engineering at Birmingham City University, UK.
6http://dmtlab.bcu.ac.uk/nickjillings/SAC/
or sends. Plugins were disabled, nor could any extra audio be imported. To investigate the ways in which subjects converge to predetermined mix settings given the starting position of their faders, each song had two randomised starting position for the volume and panning positions. The gains were set randomly to values be between -40dB and +10dB to ensure that all the tracks could be heard on the start of the session.

4 Results

Table 1 shows a list of sessions that exceeded an action threshold from the 71 total sessions submitted during the experiment. The threshold was set when the total action count was greater than double the track count (2

interactions per channel), as the action count should be relational to the users’ effort. This was used to remove sessions where people had not invested a significant amount of effort into the task. Each session took on average between 8 and 16 minutes to complete with 100 to 150 actions per session, demonstrating the balance mix is a complex task with a user interaction occurring every 4 to 10 seconds.

To illustrate the average gain structures used by participants, Figure 5 shows the relative loudness of each track against the overall session loudness. Here, the lead vocals are consistently placed higher in the mix than the other elements, as is the relative level of the bass guitar. To highlight temporal patterns found in the data, figure 6 shows that when an action is applied to a selected channel, there is a high probability the next action will take place on one of its neighbouring channels in the mixer view. This is shown through the magnitude of the diagonal cells in the matrix often having two neighbouring regions of activity, where this also indicates a pattern of operation from left-to-right.
<table>
<thead>
<tr>
<th>Song</th>
<th>No. Tracks</th>
<th>Group</th>
<th>Participants</th>
<th>Avg. actions</th>
<th>Avg. duration (mm:ss)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I'm Alright</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>87.50</td>
<td>10:03.94</td>
</tr>
<tr>
<td>I'm Alright</td>
<td>12</td>
<td>2</td>
<td>2</td>
<td>86.00</td>
<td>05:18.67</td>
</tr>
<tr>
<td>Left Blind</td>
<td>16</td>
<td>1</td>
<td>1</td>
<td>125.00</td>
<td>07:00.56</td>
</tr>
<tr>
<td>Left Blind</td>
<td>16</td>
<td>2</td>
<td>2</td>
<td>175.50</td>
<td>30:58.36</td>
</tr>
<tr>
<td>Queens Light</td>
<td>16</td>
<td>1</td>
<td>5</td>
<td>126.60</td>
<td>59:17.89</td>
</tr>
<tr>
<td>Queens Light</td>
<td>16</td>
<td>2</td>
<td>5</td>
<td>173.20</td>
<td>18:52.22</td>
</tr>
<tr>
<td>Sleigh Ride</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>76.0</td>
<td>06:25.42</td>
</tr>
<tr>
<td>Sleigh Ride</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>51.0</td>
<td>03:06.31</td>
</tr>
<tr>
<td>The English Actor</td>
<td>17</td>
<td>1</td>
<td>1</td>
<td>115.00</td>
<td>06:28.94</td>
</tr>
<tr>
<td>The English Actor</td>
<td>17</td>
<td>2</td>
<td>2</td>
<td>139.00</td>
<td>31:11.79</td>
</tr>
</tbody>
</table>

Table 1: The summary of the final sessions after filtering, with the total overall values in bold.

Further to this, the magnitude of the actions used by a single participant also decrease over time (show in figure 7). These actions exhibit high differential changes, followed by smaller changes as participants approach the final mix.

There was little variation for the loudness or action history depending on the starting position, however Figure 8 shows the mix-space position of tracks varied across the two starting positions. This highlights commonalities across mixes, such as the kick drum and overheads generally being centrally panned, and high frequency content having higher variance across the stereo image.

This suggests that there is higher disagreement between the two groups (Synth 1 and Hammond) on the selected tracks.

Tables 2 and 3 show the grouping of tracks. The most grouped tracks were the drums, with the highest number of occurrences and average number of tracks grouped. In the 353 tracks, 204 of them were sent to a group (57.79%). Only 3 groups were subgrouped in some-way, all three for effect rather than organisation.

Table 2: The most popular groups with occurrence count and average number of tracks.

<table>
<thead>
<tr>
<th>Bus Name</th>
<th>Count</th>
<th>Avg. No. of Tracks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drums</td>
<td>10</td>
<td>6.300</td>
</tr>
<tr>
<td>Bass</td>
<td>6</td>
<td>1.166</td>
</tr>
<tr>
<td>Vox</td>
<td>5</td>
<td>2.800</td>
</tr>
<tr>
<td>Kit</td>
<td>4</td>
<td>5.750</td>
</tr>
<tr>
<td>Synths</td>
<td>4</td>
<td>5.250</td>
</tr>
<tr>
<td>Keys</td>
<td>3</td>
<td>3.333</td>
</tr>
<tr>
<td>Guitar</td>
<td>3</td>
<td>2.667</td>
</tr>
</tbody>
</table>

5 Discussion

As shown in Figure 5, we are able to confirm from our mixing experiments that almost all of the mixing engineers, lead vocals have been shown to sit higher in the mix than other tracks [17, 18]. The bass guitar however, was also a prominent feature in the mix and the toms were significantly quieter, potentially due to a musical
<table>
<thead>
<tr>
<th>Track Name</th>
<th>Song name</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drums Room</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Floor Tom</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Hi Hat</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Kick</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Overheads</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Rack Tom</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Snare Down</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Snare Up</td>
<td>Left Blind</td>
<td>1.000</td>
</tr>
<tr>
<td>Overheads</td>
<td>The English Actor</td>
<td>1.000</td>
</tr>
<tr>
<td>Room</td>
<td>The English Actor</td>
<td>1.000</td>
</tr>
<tr>
<td>Snare Down</td>
<td>The English Actor</td>
<td>1.000</td>
</tr>
<tr>
<td>Snare Up</td>
<td>The English Actor</td>
<td>1.000</td>
</tr>
<tr>
<td>Hi Hat</td>
<td>Queens Light</td>
<td>0.700</td>
</tr>
<tr>
<td>Kick In</td>
<td>Queens Light</td>
<td>0.700</td>
</tr>
<tr>
<td>Kick Out</td>
<td>Queens Light</td>
<td>0.700</td>
</tr>
</tbody>
</table>

Table 3: The most grouped tracks with the percentage times grouped.

aspect of the song being produced. A large number of automatic mixing tasks work on the premise that tracks should have equal loudness for automatic mixing (e.g. [3]), however this does seem to be a common mixing procedure based on our results.

Interestingly, we found that for a song with no vocals, there was no clear ‘dominant’ lead track involved, possibly suggesting the vocals can be used as a reference when mixing. A potential explanation for the relative level of the bass guitar is that it is often made louder to emphasise the low frequency content as this is the only source that has any significant spectral energy below 100Hz.

As found in [19], the most commonly grouped instrument class was the drums, with 10 sessions in total having drums grouped together. For the most commonly completed song, Queens Light, 70% of the tracks were actually grouped, which suggests grouping trends are heavily dependent on the engineer.

The 2D mix plots in figure 8 show that in general mixing trends are followed for most tracks, regardless of the starting position of the faders (e.g. Kick In and Overheads). These types of tracks also tend to have very high levels of agreement in figure 5. However for other tracks, groupings are correlated the starting position of the faders. This indicates the starting position may influence the perception of how sound sources are related to each other.

In addition to this, figure 7 shows that as the engineer navigates the space, the magnitude of their changes gets smaller over time. Here, the initial value is classed as step 0 and the next step is after the first action is produced. The figure shows the magnitude of the deltas, between any two steps, showing the magnitude difference over time. It is clear from this that the largest magnitude changes happen earlier, which then decrease over time. This also suggests that few tracks ever needed more than 5 adjustments to their volume for their levels to converge. These findings may be useful for the development of adaptive user interfaces, in which parameter ranges can change over time for higher resolution control, when entering the minor adjustments phase of mixing.

These results will contribute to a rule-based automatic session structure, in which faders, pan positions and groups can be automatically assigned values on initialisation, based on the audio channels and semantic data from the DAW. Spectral balancing could be applied to the mix for example by amplifying low frequency components, and the vocal can be set to a prominent level relative to the other channels. Similarly, selected instruments may be centrally panned, e.g. kicks, overheads and bass, whilst others will make use of the stereo image.

6 Conclusion

In this paper we have presented a system to enhance the collection and processing of DAW session data. We have shown the online tool can capture temporal session and audio information, which is unavailable in traditional DAW software, thus allowing a deeper insight into the mixing process. We have investigated the ways in which engineers approach a balance mix and demonstrated that a large number of user interactions and decisions can be captured and utilised for session analysis. Our initial tests show that engineers exhibit common mixing trends relating to the ways in which objects are panned and vocals are mixed.

References


