StreamingLaban – Low bandwidth format for dance encoding with Labanotation

Matevž Baloh, Ciril Bohak, Matija Marolt

Laboratory of computer graphics and multimedia, Faculty of computer and information science, University of Ljubljana, Slovenia
E-mail: matevz.baloh@lgm.fri.uni-lj.si, ciril.bohak@fri.uni-lj.si

Abstract. *In the world of music, a variety of digital symbolic music representations exist, from performance oriented (e.g. MIDI) to score oriented (e.g. MusicXML). Contrary to this, in the related world of dance, while a paper friendly symbolic representation exists and is widely adopted (Labanotation or Kinetography Laban), a similar digital open standard has not yet been created. While there have been a few attempts so far, they have focused on encoding types, which require loading of all data before execution, as well as encoding more data than is necessary. Our goal is to create a format that can allow real-time encoding, decoding and interpretation of Labanotation.*

In the paper, we describe the format, its inner workings and ways of encoding and decoding various symbols.

1 Introduction

Digital symbolic music representations have a long history, as they were the only practical way of encoding music in a computer before digital audio representations became practical. Many representations exist with different purposes. For performance, the most well known is MIDI (musical instrument digital interface), which basically encodes which instrument should play which note at a given time, volume and duration [1]. Many music creation tools were developed to manipulate such symbolic representations on all user skill levels – from positioning objects on a plane to create very controlled and good sounding music with little user skills required, to beat and loop generators and sheet music editors. Because the bar to enter the field of music production has been lowered so much, we are now able to produce much more music than ever before.

Dance is tightly linked to music, yet standards for computer encoding of dance are non existent. Our goal is to produce formats, standards and software that would enable transcription, analysis and production of dance tracks. We base our work on Labanotation [2, 4], an already established dance notation standard, which however has yet to find an appropriate digital representation. We will make all developed notations and accompanying software open source and it is our hope that application developers choose to support our work.

This paper is divided into 5 sections. In Section 2 we present the basics of Labanotation. Section 3 contains an overview of related work, in Section 4 we describe our proposed dance notation format StreamingLaban and compare it to existing formats. In Section 5 we presents our findings and outline our future work plans.

2 Labanotation

Labanotation, also known as Kinetography Laban, is a dance notation created by Rudolf Laban. This form of notation is a popular way of representing and sharing dance scores among choreographers. There are many similarities between Labanotation and music notation. For example, both have staves, tracks and measures. In this respect they are compatible and a sheet of paper can contain both music and Labanotation scores, written next to each other.

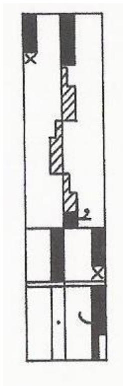

Figure 1. A small section of a Labanotation score, consisting of a staff with direction symbols and decorations.

The majority of Labanotation is defined in three levels: staff, Laban symbol and decoration. The score begins with a vertical staff with three lines. These lines define 9 vertical tracks, each for a part of the human body: four for the left side and four for the right side. Each of these sets of four contains a track for that side's foot, leg, upper body and arm. The ninth track is for the head.

Each track can contain Laban symbols. There are many kinds of symbols, however the most common symbol at this level is the direction symbol. The idea behind the direction symbol is to determine what position the effected body part is supposed to end in. For example, if the symbol for "right" is in the track for the right arm, this means the right arm is supposed to be extended to the right by the end of the Laban symbol.

Each Laban symbol can also have decorations. A very large number of decorations exist, and they define the specifics of how a move is meant to be executed. For example, while direction symbols only point in nine directions, including center, and three height levels for a total of 27 possible directions, the addition of pin decorations allows for more detailed direction definitions. Decorations can also define whether moves are meant to be performed quickly, slowly, whether they are emphasized or only implied, etc. and can also be used to more clearly define which part of the body a Laban symbol applies to. Some decorations also link several symbols together, however these are rarer.

3 Related work

The review of literature revealed that several Labanotation editing programs had been developed, however most are no longer supported or distributed. A summary work [3] mentions the existence of a number of editors: Calaban, Labanatory, Labanotation LED, MacBenesh, Benesh Notation Editor, however none of the websites listed in the paper exist today.

LabanWriter is the most widely used editor [3]. This editing software is still available and distributed via the Ohio State University Department of Dance. It is however only available as a dekstop application for the Mac, essentially a themed drawing program. The program presents the user with a blank canvas, which can then be filled with the various symbols and decorations of Labanotation. There are some limitations to what a user can draw, which stem from the rules of Labanotation. Some assistance is also offered in the form of snapping to a grid and symbols only being scalable in the vertical direction, however the tool does not give much consideration to the meaning or context of each of the symbols put on the canvas past their icon.

LabanWriter saves the notation in a proprietary text format, where each line represents a symbol, described with an identifier and its attributes such as position, height, width, etc. Apart from the symbol identifier, none of the stored data has Labanotation specific information. For example, we only know the X and Y coordinates of a symbol on a page, we do not know which staff, track or measure (time slot) the symbol belongs to. Therefore the format is suitable only for display purposes, but not for interpretation and visualization.

Led and Lintel were two programs developed at the University of Sydney, however they are not distributed anymore. Similarly to LabanWriter, the Led and Lintel programs feature a set of Labanotation related graphics symbols, which can be positioned on staves. Scores created in this way can then be printed, saved in a proprietary file format or exported into an animation format. The latter is interesting, as it can be visualized, however it only describes the movement of ellipsoids which visualize the encoded motions, but without being linked to the Labanotations itself - an animation therefore loses any relation to its symbolic description. It is also very limiting as it only allows for animation of directional Laban symbols and ignores decorations.

LabanDancer is another noteworthy piece of software [5], which claims to read LabanWriter files and animate them in a manner similar to Led and Lintel. We were however unable to find this software anywhere.

LabanEditor [6, 7], which also isn't available anymore, used to be a basic editor for Labanotation, however it is important as it stored Laban symbols not just as graphical symbols, but also with respect to Labanotation grammar. LabanEditor stores the notation in a chronological order with respect to the type of Laban symbol, the staff, duration, track, etc. Its limitations, as presented in [6] are that it can only store directional symbols. While this can easily be expanded to include other Laban symbols, the biggest flaw is it does not allow for Laban decorations, which can be tied to symbols. The nature of the format, however makes it a good starting point for our developments.

Three additional XML based file formats are important for our work: MusicXML, LabanXML and MovementXML.

MusicXML was defined by Michael Good in 2003 [8] and is actively being developed. The idea behind it is to be able to represent symbolic music in XML form. LabanXML is heavily inspired by MusicXML. It is, as the name suggests, intended to represent Labanotation scores in an XML based representation. This format is the first hierarchical format we have looked at, which is capable of representing staves, tracks, symbols and decorations in a way which makes it clear which decoration belongs to which symbol, which symbol belongs to which track and which track belongs to which staff.

Unfortunately LabanXML, as defined in [9], cannot represent the entirety of Labanotation. It is an upgrade from the formats discussed thus far, however retains some of the image parameters and additionally keeps certain elements of music, which are not common in Labanotation. Its upgrade is MovementXML [10], which extends LabanXML to represent the entirity of Labanotation, however this makes it very complex and much less human readable. The XML schema is provided and an example application (Tinikling Project) is mentioned in the the dissertation, however it is not available anymore.

For our purposes LabanXML and MovementXML both provide imperfect solutions. XML does not require children of an element to be listed in any particular order, thus there is no guarantee that, for example, an element for measure 4 will be followed by an element for measure 5. The consequence of this is that the whole XML file needs to be read ahead of time and stored in memory, thus making the format inappropriate for streaming. The other major problem is file size, which we would like to minimize. XML is text based and contains a lot of unneeded data, including spelled out element and attribute names.

4 StreamingLaban

In this section, we propose a new digital Labanotation representation, named StreamingLaban. It focuses on being small in size and contains records that are chronologically ordered and can thus be executed in real time, as they are read, without the need of knowing the entire score.

The basis for StreamingLaban is a text-based file format, which, when binarized, becomes small in size and appropriate for streaming. StreamingLaban contains two broad states: initialization and performance. During the initialization state, the data sent relates to the definition of the score, dancers, stage, static and dynamic environment, etc. This state consists of the following parts:

- score information,
- performance parameters,
- stage,
- props,
- actors,
- groups,
- staves.

The score information contains the score name, author and other score related metadata. This section is inspired by ID3 with consideration given to tags being dance related, as opposed to music related.
Performance parameters are system level parameters which relate to the start of the score. These are currently taken from LabanXML and include the beat and beat type.

The stage parameters are taken from Led and Lintel format. These are the width and depth of the rectangle which represents the stage. At this time we are only foreseeing rectangular stages, which can be adjusted via the next part: props.

Props are objects which cannot be animated through Labanotation, but can be interacted with. This typically represents objects placed in the world. All props have a globally accessible and unique ID, which can be used in the Laban score. For example if an actor is meant to touch an object or carry an object, they can use the object's ID to define this.

Actors are humanoid objects in the Laban score. Labanotation is only applicable to humanoid actors, which we assume to be the only category of dancers able to perform a score. It is also noteworthy that humanoids with missing limbs can be animated with Labanotation, the only potential issue is that humanoids with extra limbs cannot be animated with it. Much like props, actors also have unique IDs, which can be referenced in the Laban score.

A Laban staff can give instructions to an entire group instead of an individual actor. This is why actors can be grouped. Groups contain the arrangement (circle, phalanx, inverted circle, etc.) and a list of actors which make up the group. Groups also have their own unique ID, which can be referenced in the Laban score.

The last part of the initialization process is the definition of staves. Each staff is tied to an actor or group. Staves also contain an ID which identifies which staff a given Laban symbol belongs to.

It is also worth mentioning that most of these settings can be changed during the performance state. This means a valid score can contain nearly nothing in the initialization state and all the actual initialization as part of the first measure in the performance state. The benefit a separate initialization state is that the initialization makes up the bulk of an i-frame's content. If we are streaming a StreamingLaban encoding performance, a user who connects in the middle of the stream only requires an up-to-date i-frame to begin Labanotation playback from that point without a need to play everything from the beginning up to that point. This means that maintaining an up-to-date initialization state during performance makes it easy to create i-frames or to initialize a newly connecting client. The full i-frame would however also contain the current state of the actors and props, which is defined in the performance state.

Performance data follow the initialization stage. The state is listed chronologically from time slot -1 on. Time is divided into discrete segments, with a maximum resolution of a sixteenth note (semiquavers). Time slot -1 is the state of the actors at the beginning of the performance. This time slot is not animated, the end state of the animation is simply applied before the performance begins.

Time slot 0 then begins the performance in sixteenth note increments. Each time slot needs to be received in full for its contents to be animated. The time slot contains Laban symbols which begin at that time slot. Each Laban symbol contains the parameters which define it (including type, staff, track and duration, all explained in Section 2), as well as a list of decorations which better define how the symbol is meant to be interpreted. The way this is done is by first including a line which contain a Laban symbol with a defined ID and one or more subsequent lines with a decoration which gets tied to the symbol's ID. This method works because decorations never impact several time-slots. Even lines which connect multiple Laban symbols only do so horizontally or vertically, never diagonally. Horizontal connections apply to the same time slot while vertical connections only mean the two vertically stacked symbols are meant to be performed simultaneously, which in practice means they start in the same time slot. This means that Laban symbol IDs only need to exist within the time slot and can be discarded as soon as it ends. This is important because it means there are no dependencies between time slots and initialization for new users will not need rely on previous Laban symbol IDs.

Along with Laban symbols and decorations, system messages can also be included in the performance, which serve to create or remove staves, groups or actors. The idea is that all unused assets are pruned at the end of a time slot. So if an actor is tied to a terminated staff at the end of a time slot, and not tied to another staff or a used group, then that actor is pruned and needs to be recreated before they can re-enter the performance.

5 Conclusion and further work

In this paper we outlined the idea behind our StreamingLaban digital dance format. The main strength of this file format is that it encodes Labanotation as opposed to an image representation of Labanotation, it is small in size, is hierarchical and it can be interpreted in real time without needing the whole score.

The main current drawback of the proposed format is the decision to only allow for one level of decorations under a symbol – a decoration cannot have decorations of its own. This, however can be amended in our future work. Our future work includes the definition of specific details that make up the file format – the keywords and parameter order, a validator, interpreter, editor and dance visualizer.

References

[1] MIDI Tech Specs and Info, http://www.midi.org/techspecs/

[2] A. H. Guest: Labanotation – The system of analyzing and recording movement, Taylor and Francis, 1977

[3] T. Calvert, L. Wilke, R. Ryman, I. Fox: Applications of Computers to Dance, March/April 2005, IEEE Computer Graphics and Applications

[4] B. Ravnikar: Kinetografija, ples in gib, Zveza ljudskih in tradicionalnih skupin Slovenije, 2004

[5] L. Wilke, T. Calvert, R. Ryman, I. Fox: From Dance Notation to Human Animation: The LabanDancer Project, 2005, Computer Animation and Virtual Worlds, volume 16

[6] K. Kojima, K. Hachimura, M. Nakamura: LabanEditor: Graphical Editor for Dance Notation, 2002, IEEE Int. Workshop on Robot and Human Interactive Communication

[7] W. Choensawat, S. Takahashi, M. Nakamura, K. Hachimura: A Labanotation Editing Tool for Description and Reproduction of Stylized Traditional Dance Body Motion, 2011, Digital Humanities

[8] M. Good, G. Actor: Using MusicXML for file interchange, 2003, Web Delivering of Music 2003

[9] M. Nakamura, K. Hachimura: An XML representation of labanotation, LabanXML, and its implementation on the notation editor LabanEditor2, 2006, Преглед НЦД

[10] J. Hatol: MovementXML: A representation of semantics of human movement based on Labanotation, 2006, Simon Fraser Universiry