A Generic Approach to Design and Querying of Multi-Purpose Human Motion Database

Wiktor Filipowicz[1], Piotr Habela[1], Krzysztof Kaczmarski[2] and Marek Kulbacki[1]

[1] Polish-Japanese Institute of Information Technology, Warsaw, Poland
[2] Warsaw University of Technology, Warsaw, Poland
{wiktor, habela, kulbacki}@pjwstk.edu.pl, k.kaczmarski@mini.pw.edu.pl

Abstract. The advancement of motion recording, analysis and synthesis techniques, together with the standardization of respective data formats, constitutes a solid foundation for research based on motion data sets in the context of various disciplines. However, most of the motion data sets available offer groups of files as acquired from the motion capture systems. The problem with such data is that it usually represents a single viewpoint, its context varies and is more or less implicit, and the lack of functionality implemented atop of such data set limits the data analysis and search potential. This encourages us to look at this problem domain from the database management systems (DBMS) state of the art point of view. In this paper, we outline some important aspects of applying a DBMS to motion data with the aim to provide a highly universal, extensible, shareable and searchable resource. To avoid being locked into a specific area of application, we take a very abstract view of the data and attempt to assure, a versatility and genericity of the resulting system.

Keywords: motion capture, multimedia databases, structured data, extendibility, database management

1 Challenges for a Motion Data Analysis System

Human motion databases (HMDB) have become very popular, gathering large amounts of often synchronized motion capture streams and other motion related data [1]. Captured motion data sets have multiple attributes and dimensions and different lengths for even similar motions. There is no direct correspondence between data of two different motions. The challenges in HMDB include feature selection relevant for motion comparison, indexing methods for fast retrieval of preferred motions, cleaning and compressing data sets, identification of base characteristics of human motion especially of outliers. To classify and recognize motion data correctly we often represent each motion stream as feature vector. Query formulation by feature extraction gives only statistically accurate results based on the frame rate or the position of the motion stream but rare on exact object matching. Nevertheless initial preprocessing and generalization by reduced representations can dramatically accelerate retrieval times. Forbes and

Fiume [2] relying on dynamic time warping (DTW) demonstrate how clustering and dimensionality reduction can improve retrieval time. To the same effect Chiu et al. [3] use self organizing map (SOM) for clustering and separately index segments of the body to improve retrieval times. For retrieval of motions from examples in iterative way Kovar and Gleicher [4] preprocessed locally optimal time alignments of motions for whole HMDB. Muller et al. [5] employ binary features to represent various poses and demonstrate how to produce radically faster motion retrieval. Quoted retrieval examples represent a group of retrieval problems where complete example motion is provided as a query. Query based methods in multimedia databases are not able to represent user's subjective intentions perfectly. Second group of retrieval problems in HMDB require from user browsing through database for expected motion. Liu Ren [6] explores representation for easier data visualization in browsing-style interface. Assa et al. [7] present a visual synopsis of motions for elegant database browsing. Sakamoto and colleagues [8] demonstrate visual interface for motion selection by identifying key frames from posture map obtained from SOM algorithm. Third group of retrieval problems relies on retrieval from small set of controls to capture user's intention more reliably. Liu et al. [9] propose hierarchical database comprised of local linear models for fast retrieval of a character pose from a sparse marker configuration.

Motion stream as relatively new type of multimedia require considering temporal and spatial multiatribute motion nature. This variability of data processing requirements demands for a a large degree of flexibility in underlying data management. At the highest level we recognize four main groups of requirements for the motion database system:

– to offer efficient, reliable, flexible and robust data storage;
– to allow querying, analysis and data retrieval;
– to organize cooperation of different groups of users;
– to enable the extensibility features.

Each of them influences the others and therefore should always be discussed together. An ideal system allows users to store any data related to human motion domain: database records, files of any type and size. The data and files are searchable in different ways, by built-in tools and also by custom procedures and filters. The system guarantees data protection. Hence, to enable cooperation, it should able to anonymize data to assure privacy of captured subjects. An ideal system gathers groups of users and allows them to share not only data but also applications working on the stored data. Community-developed tools for motion analysis can be shared and executed by many users in different situations. This kind of system is similar to community web portals where all content can be shared and is in fact a product of a common development effort.

2 Existing Examples of Public Motion Data Sets

We investigated publicly available data sets including the most important and well known [1], [10], [11], [12]. In all cases the phrase "database" means that there is data that can be accessed (read) by users. Only one of them facilitates uploading of data using a web interface. Sharing and community interactions are therefore usually impossible. All those datasets offered data stored as single or grouped files containing video, motion and also other synchronized data.

Large motion data collections can also constitute a challenge in the medical field. Simple tools for motion data acquisition and processing may fall short when mining in the existing archive is considered. Motion capture lab workstations are clearly not capable of storing massive amounts of records for an extended period of time. A hospital based motion capture lab which we analyzed is forced to frequently archive its motion data using DVD media. Hundreds of disks are stored and an index of their content in terms of patient data location is maintained. Although patient records become documented, the research potential of such a big data set becomes void due to poor search and analysis capability. A dedicated data server with huge capacity could definitely help, but only if files are stored under a professional database system control instead of raw files.

Animeeple [13] is a character animation tool which can import 3D files and publish characters and animations for online sharing or sale. A number of popular formats is supported by its import and export features. The available characters and animations can be combined from different sources. Content can be shared for free or sold by registered sellers and profiles are used to share information about individual content providers. As a 3D modeling and animation tool, Animeeple aims to be an easy to use application targeted mostly at less advanced users and anyone interested in publishing their content for others to use.

Gaitabase [14] is a database of medical records and data captured by gait analysis systems with a web interface. It includes a publicly shared sample data set which can be used as part of a demo. The main purpose of the system is to store gait data files, provide query mechanisms and display the results for statistical analysis. Gaitabase is a relational database with tables storing spatial-temporal files. The search capabilities offer a way to define filters based on search criteria with multiple conditions and logical operators. Processed filters return records from the table of patients as search results. The analysis functionality built into the system makes it possible to plot gait data, select variables and compare sets of data.

As can be seen, there are significant limitations of storing motion data set files "as is", without a dedicated data management tool. Then we could observe how diverse the expected features of such a tool can be depending on the area of application. For multimedia authoring, the identification of motion kind, interoperability in terms of different formats provided and capability to edit motion are essential. For the medical applications the primary subject of queries are the attributes describing a patient and data processing is a more read-only fashion. In both cases efficient ways of querying and easy access to retrieved data stored inside motion data files are desirable.

3 Required Features of the Database

Useful features, high efficiency and stability, place the DBMSs in general right in the center of our attention. Confronting our requirements with the features traditionally associated with multimedia data management [15], we need to consider the following:

- Handling large, raw data files,
- Need for automatic metadata extraction for imported items,
- Some aspects of temporal querying,
- Need for custom query optimizations (including specific indexing),
- Specific presentation tools and user interface design.

On the other hand, contrary to some other multimedia applications, no real-time challenge occurs. Moreover, the spatial aspect of data – although it obviously applies, will probably not be handled explicitly by the database mechanism itself. Moreover, a specific challenge is the need for high degree of universality and openness for new data formats, metadata elements, search criteria, transformation and analysis routines, connected with the ability of secured shared access and resource contribution.

Data storage and querying

Despite the amount of information is high, the system's core data elements are not expected to form big graphs and would not be the subject of intensive navigation (i.e. much of the complexity would remain opaque to the database schema structures). The foundational set of entities is considered to form a following hierarchy: **Performer** – identifies a subject of motion; either a human or an individual created using 3D modeling software; **Session** – identifies motion capture sessions for a single performer; **Trial** – identifies trials which may be done by a performer during a single session; **Segment** – describes a fragment of the trial sequence distinguished based on some criteria of interest. Different kinds of segmentation of a single segment are possible, especially facing the multiple areas of applications. At each level of this hierarchy it should be possible to assign instances to categories and to equip them with custom attributes. Good support for bulk data is needed to assure fast and reliable motion data storage and retrieval as most of the data will remain stored inside such large objects. One of the most important factors for data querying is the intuitiveness of the data structure as perceived by the database user. From this point of view the following elements need to be covered with a suitable abstraction:

- Links between categories, performers, sessions, trials and files to ease and simplify the formulation of queries referring to more than one of those entities;
- Named generic attributes and their values should be as easy for browsing and querying as regular, statically defined features;
- Features that can be determined by processing motion data files (here called "analytical features") should be available just like plain, stored attributes;
- Services providing derived data that describe certain entity should also be available in a similar way as regular attributes.

The above assumptions result in a concept of object-like, mostly hierarchical structure, having a significant amount of behavior is built in and where the distinction between attributes and operations may be hidden from the user.

Extensibility features and user cooperation
When analyzing extensibility we must focus on the two main aspects of the system: data storage and data querying. In case of the first one we expect that the data stored in the system does not necessarily have to be limited to predefined attributes or file formats. Therefore, no limitation on file type is acceptable. Similarly, a fixed set of attributes for entities is just a starting point from which particular users may start building their own information representation. Summing up, the contributable features would include custom attributes, named queries (filters) and behavior extending data analysis and import / export capabilities. Those behavioral extensions require the following:

1. Standard interfaces to existing data and calculable features, which will be used by new components;
2. Connection points for new components and their safe execution control;
3. A descriptor file specifying the semantics and configuration.

Filters, being a more lightweight construct, only require designing appropriate syntax assuring easy retrieval of existing filters for invocation as well as for modification. For extended cooperation, the system should allow to distribute common activities, including the creation of behavioral parts (e.g. queries or analytical features). This requires many parts traditionally perceived as second class citizens of the system to be promoted to first class citizens. This calls for a new paradigm for user interfaces, query processing and security. These issues are broader discussed in the next section. Security is perceived as an important requirement in the system, especially where medical records are involved. The privileges may need to be differentiated both vertically (against the types of data) and horizontally (particular subsets of data). Hence the access rights pattern goes beyond simple data ownership criteria and would need to be controlled by the application. At the same time, for flexibility and performance, the database content cannot be encapsulated too aggressively against the behavioral extensions.

4 Extensible Architecture for Database and its Services

Reconciling the genericity and approachability requirements makes it necessary to establish three levels of abstraction:

– **implementation**, including the implementation of internal structures for generic features and resource representations as well as access optimization;
– **design**, outlined by programming and configuration interfaces for administrator and extension developers;
– **use**, where built-in and extension features, both stored and calculated are provided in a uniform way for querying and retrieval.

Here we focus on the second level, perceiving it pivotal for the overall design.

Conceptual view of the generic data
The features considered led us to the following conceptual design, expressed in fig. 1 as a UML class diagram. Note that its complexity would be hidden from regular user, who would actually deal with quite straightforward hierarchy of the resource data.

The foundation is formed by the core conceptual entities to be presented to the user: the Performer…Segment hierarchy. We call them Resources. Two of them: Trial and Segment possess temporal characteristics, that are to be supported by queries and when extracting temporal fragments of those resources. The synchronization between different representations would be the substantial value of the system (allowing to investigate the relationships between particular channels). Apart from the resources, we need to explicitly deal with their Representations. Although ideally the representations could be perhaps transparent to this conceptual structure, in fact we need to include them also as the subject of extensible metadata mechanism to support various formats, levels of precision etc. The extension mechanism makes the structure open to new data types in two main roles:

- Item metadata attribute sets - establishing virtual subclasses for resource, whose visibility can be differentiated on a per user basis. For simpler classification a similar, more intuitive notion of Item Group is also available.
- Resource representations - allow supporting new types of stored files or calculated, derived documents.

Behavior for extendible features
The data content extensions of the database entail the need for dynamically incorporating respective behavior:

- New areas of application may bring not only new stored attributes, but also new calculable features. Though made available to the user in a uniform way with the stored attributes, they need a way to connect their calculation algorithms to the query evaluation and reporting routines.
- New resource representations would usually require specifying import behavior that would be capable of automatically extracting their content-dependent metadata and writing it explicitly into the database.
- If a given resource representation is not materialized, it will need submitting an algorithm of producing it based on other resource data (export behavior).
- New topics of research can produce recurring query predicates, so naming and storing them could be useful for sharing or reuse. This requires a relatively simple expression language that could be represented by database-stored Filter entities.

The presence of computation-intense virtual features and derived resource representation makes it also necessary to introduce the postponed result collection query and data retrieval API operation variants.

Plug-ins and sandboxes
Plug-in based extensibility is a commonly accepted approach for enriching software functionality. However, this is mostly used for desktop or server applications under strict administrator control. A user granted with administrator privileges

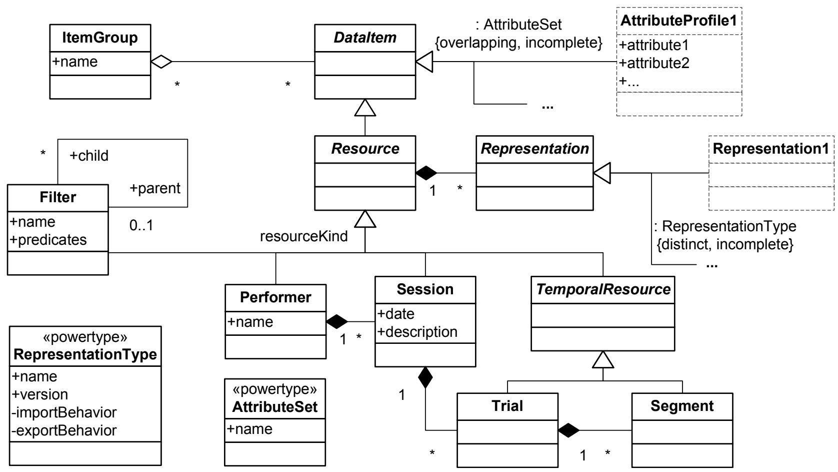

Fig. 1. Motion database structure as perceived by its user

is responsible for any problems caused by a faulty plug-in. Also intranet users pose less risk compared to global network users. If a community open server is to allow users to execute their own code, it must assure a proper level of security:

1. No data can be seen by plug-ins which are not granted with certain privileges;
2. No data or server application can be destroyed by a plug-in code;
3. A plug-in cannot directly or indirectly influence execution of other plug-ins;
4. No plug-in can consume server resources beyond a given level;
5. No plug-in can cutoff normal server communication lines.

Therefore, we consider guarding each executed plug-in in a sandbox, which will be controlled by dedicated server processes. A sandbox-based control means that a process inside is fairly free to perform different tasks, create other processes, data files, etc. if only it obeys given rules (for example, disk occupation or processor load). If a process is trying to break from the shelter it is immediately terminated. Removing its sandbox includes deleting of any intermediate data, rolling-back transactions, etc. The server must offer appropriate interfaces for defining new plug-ins, configuring them, sharing code among other users, controlling their execution, retrieving results. A system similar to [16] could be used for these tasks.

5 Conclusions

We presented here a vision of a generic motion capture database system offering extended features like data sharing and user cooperation, server-side query processing and analysis of motion data, user-defined plug-ins, generic custom defined attributes and openness for a community of different types of users. There has been no similar system created so far and we believe that implementation and deployment of this solution will influence human motion capture society. To realize this vision, several aspects of transparency need to be achieved to provide users and developers with:

– generic metadata attributes which are as simple in querying as the statically defined ones,
– calculable metadata and resources accessible like stored data,
– access rights control factored out of the plug-ins code,
– resources management protecting the performance and availability.

The system seems to be complicated at the server side and since it is a scientific solution, at the beginning it will be implemented in an inventive and incremental way. Small portions will be published for a closed group of users who will test it, collect remarks and help to define the next steps. The process of polishing all requirements and enabling high level of security and reliability will take years, but already the partial results might be useful for large groups of users.

Acknowledgments

This paper has been supported by the project ,,System with a library of modules for advanced analysis and an interactive synthesis of human motion " co-financed by the European Regional Development Fund under the Innovative Economy Operational Programme - Priority Axis 1. Research and development of modern technologies, measure 1.3.1 Development projects.

References

1. L. Sigal, A. Balan, and M. Black, "Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," IJCV, vol. 87, no. 1, pp. 4-27, March 2010.
2. Kevin Forbes and Eugene Fiume.: An efficient search algorithm for motion data using weighted pca. In 2005 ACM SIGGRAPH/Eurographics SCA, pages 67-76, July 2005.
3. C. Y. Chiu, S. P. Chao, M. Y. Wu, and S. N. Yang.: Efficient content-based retrieval for motion capture data. Journal of Visual Communication and Image Representation, 15:446-466, 2004.
4. Lucas Kovar and Michael Gleicher.: Automated extraction and parameterization of motions in large data sets. ACM Transactions on Graphics, 23(3):559-568, August 2004.
5. B. Demuth, M. Müller, and B. Eberhardt.: An information retrieval system for motion capture data.In Proceedings of ECIR, pages 373-384, 2006.
6. Liu Ren. Statistical Analysis of Natural Human Motion for Animation. PhD thesis, CMU, 2007.
7. Jackie Assa, Yaron Caspi, and Daniel Cohen-Or.: Action synopsis: pose selection and illustration. ACM Transactions on Graphics, 24(3):667-676, August 2005.
8. Yasuhiko Sakamoto, Shigeru Kuriyama, and Toyohisa Kaneko.: Motion map: image-based retrieval and segmentation of motion data. In 2004 ACM SIGGRAPH/Eurographics SCA, pages 259-266, July 2004.

9. Guodong Liu, Jingdan Zhang, Wei Wang, and Leonard McMillan.: A system for analyzing and indexing human-motion databases. In Proceedings of the ACM SIGMOD, pages 924-926, 2005.
10. Gaeorgia Tech: HID Database, http://www.cc.gatech.edu/cpl/projects/hid
11. CMU: Carnegie-Mellon Mocap Database, http://mocap.cs.cmu.edu
12. CASIA Gait Database, http://www.sinobiometrics.com
13. Animeeple Animation Software, http://www.animeeple.com
14. GAITABASE, http://gaitabase.rch.org.au
15. B. Prabhakaran. Multimedia Database Management Systems, Springer 1996
16. Global Alliance WS-Resource Framework, http://www.globus.org/wsrf