****

**INNOVATIVE ECONOMY OPERATIONAL PROGRAMME**

**A system with a library of modules for advanced analysis and interactive synthesis of human figure movement**

**Motion**

|  |  |
| --- | --- |
| **Report 2.4** | **Assumptions of the design of the database oriented  towards the collection of character movement data** |

|  |  |
| --- | --- |
| **Current Version Date:** | 2009-10-05 |
| **Editors:** | Marek Kulbacki, Piotr Habela, Paweł Lenkiewicz, Krzysztof Kaczmarski |
| **Document Location** | [http://ruch.bytom.pjwstk.edu.pl](http://ruch.bytom.pjwstk.edu.pl/) |
| **Version** | 1.02 |

# Abstract:

|  |
| --- |
| The "Character Movement Database" component, also referred to here as the "Database System" (SBD), will be used to permanently store the motion sequences recorded during the measurement sessions, the various session and actor data entered with them, and information related to the classification and segmentation of the recorded movement. Its task will be to ensure the storage and retrieval of motion data – both for the specific needs of domain services and as generic data, potentially useful in other applications.  This document presents the assumptions regarding the design work made on the basis of the stage of recognizing the state of the art. It is the starting point for a detailed analysis carried out in parallel with the implementation work and based on partial results of the remaining ongoing project tasks and information provided by domain experts. |

Table of contents

[Abstract: 2](#_Toc257624439)

[1 Introduction 5](#_Toc257624440)

[1.1 Purpose of this document 5](#_Toc257624441)

[1.2 Functionality Overview 5](#_Toc257624442)

[1.3 Document Structure 5](#_Toc257624443)

[2 Characteristics of major motion standards and datasets 6](#_Toc257624444)

[2.1 Motion description formats 6](#_Toc257624445)

[2.1.1 Acclaim ASF / AMC 6](#_Toc257624446)

[2.1.2 Collada DAE 7](#_Toc257624447)

[2.1.3 Autodesk FBX 8](#_Toc257624448)

[2.1.4 C3D 8](#_Toc257624449)

[2.1.5 Biovision BVH 9](#_Toc257624450)

[2.1.6 VSK/V 9](#_Toc257624451)

[2.2 Publicly available datasets 10](#_Toc257624452)

[2.2.1 HumanEva I 10](#_Toc257624453)

[2.2.2 HumanEva II 11](#_Toc257624454)

[2.2.3 Gatech HID 11](#_Toc257624455)

[2.2.4 HDM5 11](#_Toc257624456)

[2.2.5 KUG 12](#_Toc257624457)

[2.2.6 CASIA Gait 12](#_Toc257624458)

[2.2.7 CMU MoBo 12](#_Toc257624459)

[2.2.8 CMU-MMAC 12](#_Toc257624460)

[3 Context – list of modules 14](#_Toc257624461)

[4 Component Functionality Specification 15](#_Toc257624462)

[4.1 External actors 20](#_Toc257624463)

[4.1.1 Users 20](#_Toc257624464)

[4.1.2 Cooperating Components 20](#_Toc257624465)

[4.2 Use Case Specification 21](#_Toc257624466)

[4.3 Key Non-Functional Requirements 23](#_Toc257624467)

[5 Database schema 24](#_Toc257624468)

[5.1 Initial schema design for the database 25](#_Toc257624469)

[5.1.1 Introductory information 25](#_Toc257624470)

[5.1.2 Diagram sketch for the identified elements of the problem domain 25](#_Toc257624471)

[5.2 Assumptions for detailed design 29](#_Toc257624472)

[6 Database Component Interfaces 31](#_Toc257624473)

[6.1 Visual Interface Concepts 31](#_Toc257624474)

[6.2 Application Interfaces 33](#_Toc257624475)

[7 Technical assumptions 34](#_Toc257624476)

[7.1 Volume of data considered 34](#_Toc257624477)

[7.2 Choosing a Database Management System 35](#_Toc257624478)

[7.3 Server Hardware Requirements 36](#_Toc257624479)

[7.4 Operating instructions 36](#_Toc257624480)

[7.5 Plugin-based architecture postulate 36](#_Toc257624481)

[8 Summary 37](#_Toc257624482)

[9 Glossary of terms 38](#_Toc257624483)

[10 Origin 40](#_Toc257624484)

[11 List of illustrations 42](#_Toc257624485)

# Introduction

## Purpose of this document

Due to the research nature of the work and the need to experiment with implementation solutions, design and programming work is carried out in an iterative manner, allowing for a significant degree of evolution of the original solution. This approach is reflected in the very structure of the project's tasks, which explicitly distinguishes the initial and final versions of the software as part of the implementation work. In addition, in the design phase of work, there is a need to establish a common reference point for individual research and implementation areas as early as possible. From these assumptions, the task of the motion database project was placed quite early. The purpose of this document is to reflect and transfer to the design level the original assumptions as to the functionality of the project, enriched with the results of the analysis of methods, techniques and standards. The database design, after additional consideration of early results from musculoskeletal model design, condensation algorithm, pre-processing algorithms, clustering and motion analysis, will then be reviewed and critiqued by the groups that make up the above algorithms and, after necessary updates, will be used to implement the database component with the necessary interfaces. It is assumed that in the first phase of the project, the structures of storing the underlying data will be determined, and only with the development of research and design work focused on character models and detailed applications of motion analysis will the structures of interpreted data be specified. Hence, in this document, we focus on the first of the above-mentioned areas.

## Functionality Overview

The "Character Movement Database" (BDR) component, also referred to here as the "Database component", will be used to permanently store the motion sequences recorded during the measurement sessions, the various session and actor data entered with them, and information related to the classification and segmentation of the recorded movement. The data will be recorded in a raw form - consistent with the one obtained from the recording equipment, and in a processed form - selected and specified in accordance with the intended area of application. The system will also offer the necessary interfaces for components that process motion data and offer selected domain functionality. Finally, the BDR component will include a remote access interface that allows you to search and retrieve generic data based on preset classification criteria and quantitative parameters.

## Document Structure

The document presents the assumptions adopted during the design and implementation of the Motion Database component. Chapter 2 presents the most important formats for recording motion data and representative motion data sets that can serve as inspiration for database design solutions. Chapter 3 provides information about the architectural context of the component. Chapters 4 and 5 define the functionality of the component and the database schema, respectively. They will be updated as the detailed design progresses during the implementation phase. Chapter 6 introduces the concept of the visual interface. Chapter 7 describes the technical assumptions for the implementation of the database component. Chapter 8 provides a summary.

# Characteristics of major motion standards and datasets

Publicly available data sets and popular 3D formats are an important reference point for designing the data structure for a motion database. This section presents the most relevant such sources.

## Motion description formats

### Acclaim ASF / AMC

A file format for recording character movement based on a skeletal model, originating in the field of computer games [MRCE05][Wisc99a]. Popularized thanks to its adoption as an output format from the Vicon system software.

With the discontinuation of Acclaim, the format is no longer supported, but there are significant resources of motion sets and the functionality to export to this format from popular tools – e.g. Vicon BodyBuilder.

The format is based on two types of ASCII text documents (which can have a one-to-many relationship):

* ASF (Acclaim Skeleton File) – information about the skeleton. The definition is bone-based (as opposed to BVH, joint-based).
* AMC (Acclaim Motion Capture) – a record of the character's movement, consistent with the skeleton defined for it.

The structure of an ASF file distinguishes between a header, bone data, and a skeletal hierarchy.

The header specifies the units of length (in the form of a divisor for size in inches), mass, and angle.

The section describing the bones indicates the root of the skeleton, which has a different structure due to its specific features. The root specifies the order in which degrees of freedom are given in AMC files, the Euler angle convention for the root coordinate system, and its initial position (translation and orientation). The data for each bone includes an identifier, a name, a direction vector for the bone, a vector describing the neutral position of the bone relative to the global coordinate system, and the length of the bone. The direction vector multiplied by the length of the bone will determine the offset of any child elements. In addition, the axis of rotation of a given segment in relation to the parent's system is determined. There are also limits to the range of min and max angles. for each given degree of freedom of the joint in question.

The hierarchy section specifies the parent and one or more of its children on a per-row basis.

The requirement that the skeleton should contain no gaps results in more skeletal segments. For this purpose, the so-called *dummy bones are introduced*, which do not serve to animate but to maintain the continuity of the kinematic chain or fill the structure when converting between different skeletons and must be taken into account equally with other joints during calculations.

The AMC format contains a list of numbered frames, made up of rows dedicated to individual bones. Each of them begins with the name of the bone, followed by numbers indicating the rotations for the degrees of freedom available for that bone. They are formulated as relative rotations for the rotation axes of individual segments specified in the skeleton definition.

The format is human-readable, which unfortunately entails reduced parsing efficiency, especially in the face of significant redundancy. The subject of criticism is the lack of information about the associated ASF, the frame rate or the sampling rate in the AMC file format. This type of information can be stored in the "documentation" section of the ASF file.

Euler's convention for ASF/AMC files is always based on the right-to-left multiplication order.

### Collada DAE

DAE (Digital Asset Exchange) is a format created as part of the COLLADA (COLLAborative Design Activity) initiative, led by the Khronos Group, a non-profit consortium [BaFi08].

The format is based on XML and specifies a standard XML document schema for interchanging 3D information between different applications. The roots of the format are related to the field of computer games. COLLADA is also used as the native format used in Google Earth.

In the currently available versions, the format is extended with support for properties and physical constraints - concerning individual objects and global ones. It also allows you to specify coatings and material characteristics. The design assumption of the format is lossless recording. Versioning is also supported.

Version 1.5 from 2008 opens up a new area of application for CAD modeling (including: boundary representation - more efficient and lossless representation of solids compared to meshes; kinematics). The representation of complex joints is supported along with the specification of degrees of freedom.

Data structures representing objects are based on matrices, for which it is possible to define the method of stream reading (accessor element).

Animations can be defined based on keyframes, for which you specify an interpolation algorithm and an output channel (sampler and channel elements). Defined animations can be instantiated within an animation\_clip element, by using the command (element) instance\_animation.

The geometry element determines the visual shape and appearance of an object in the scene. Acceptable child elements allow you to define a shape in a variety of ways.

The rotate element specifies the rotation of an object around an axis. Contains a list of four floating-point numbers that specify the rotation vector [ X, Y, Z ] and the rotation angle in degrees.

The translate element changes the position of the object in the local coordinate system.

A node represents the object that makes up the scene. It is the root of the subgraph in the scene graph. It can specify translation and rotation.

A characteristic feature of the format are arrays (m.in. integer and float types) implemented as elements with text content separated by white space. An accessor that refers to such an array allows you to specify how (step, offset, expansion) such a string of numbers should map to a series of data defined by a set of named variables.

The data flow in COLLADA documents consists of the source and input elements, where the former can contain raw data devoid of semantics, while the latter specifies its semantics, in order to use the data in the definition of the parent element for input - which is therefore the consumer of the data flow. The input definition allows you to reduce data redundancy by assigning the same items in the data series to more than one attribute.

Import and export of COLLADA documents is supported by the open-source libraries [COLLADA DOM](http://en.wikipedia.org/w/index.php?title=COLLADA_DOM&action=edit&redlink=1) and [FCollada](http://en.wikipedia.org/w/index.php?title=FCollada&action=edit&redlink=1). The COLLADA DOM interfaces are generated at compile time from the COLLADA schema. It is a low-level interface that performs document parsing. It is limited to the current version of COLLADA. FCollada from Feeling Software, offers higher-level interfaces and supports imports from all versions of COLLADA. It is used by ColladaMaya, ColladaMax, and by several commercial game engines.

It is important to note that COLLADA was designed as a data exchange format. The form of data storage (XML document) and the intensive use of references between document nodes make this format suboptimal as a working format for performance reasons.

### Autodesk FBX

3D data interchange format. It allows you to represent complex 3D scenes. A format created by Kaydara for motion capture. The format specified as object-based. It allows you to record motion data with accompanying information (m.in. audio and video). It is the native format for Autodesk MotionBuilder, which is also supported by Autodesk Maya and Autodesk 3ds Max via plug-ins.

It can take binary or ASCII form. It enables the acquisition and exchange of 3D assets. Includes, m.in *keyframes*, motion capture animations, materials and textures, lighting, cameras, inverse kinematics.

A Software Development Kit for FBX is available.

### C3D

It is an open format for recording biomechanical data (especially from 3D observations) into a single disk file [C3D2009] [MLS08]. A very important assumption of the standard is the ability to define any parameters, thanks to which the file is "self-documenting" and very flexible. The assumption is to collect in one file a complete set of data on the observation – including static data of the performer, data on the configuration of measuring devices, recording parameters, units used, as well as interpreted data. The information about physical measurements stored in the file can be of two types: 3D positional coordinates with the determination of their accuracy, and analog data associated with the same frame (e.g. indications of a dynamometer platform or sEMG measurements).

The data contained in the file can be divided into three parts: header information, measurement data and additional parameters related to this data.

Part One: Header Information. Parameters such as scale are saved here, so it is possible to transform data to real values, frame rate, number of frames.

Part Two: Parameters Section. Definitions of any parameters with a description can be found here. The parameter can apply to any measurement point. Each parameter contains a label and a description. A parameter can have several dimensions that specify how many data elements its value can hold. The values belong to one of the following types: integer, float, character, byte. Parameters can be grouped.

Part Three: Data Section. The coordinates are recorded in the form of X, Y, Z along with data on the possible measurement error and the cameras that were involved in recording this point. Values can be written as an integer or floating-point. Successive frames of measurement data are stored in the file.

### Biovision BVH

It is a textual format that describes a specific sequence of motion in the form of hierarchical motion paths. The skeleton and data are stored in a single file [Wisc99b].

The file consists of two parts: The header specifies a hierarchy that contains the initial pose and a data section that contains the movement data (the translation and rotation values according to the channels defined in the hierarchy section, with the *root* element including translation and rotation, and the other segments (usually specifying only relative rotation) performing local transformations of the kinematic chain starting at *root*). and 3 values for rotation). In the header we can see the positions of the individual bones in relation to the center of the coordinate system. In the data section, there is data about the number of frames and the sampling rate (equal to the inverse of the number of frames per second (*time frame property*)) and subsequent values of X, Y, Z offsets and rotation relative to X, Y, Z according to the definition of the degrees of freedom of individual elements of the armature. The positions of individual segments of the skeleton were determined as relative – in relation to the parent element.

### VSK/V

VSK (Vicon Skeleton File) - as the name suggests, is a format used by Vicon Motion Systems to make recorded motion data available in an interpreted form based on the definition of the skeleton [CMU09], [Vico01]. The .vsk file contains the definition of the skeletal hierarchy (free, ball and socket, 2 hinges, hinge, rigid).

The definition of a specific skeleton instance is created on the basis of a previously prepared template (VST - Vicon Skeleton Template), which defines the hierarchy of the skeleton and markers designed to determine its position. The identification of the markers is carried out interactively, based on a test recording.

The .v file contains the data collected as part of the observation. To simplify processing, a fixed set of units used in the format (radians, mm, N, Hz, kg, Nmm) was adopted.

The .v file usually contains, for each of the dice, the coordinates T-X, T-Y, T-Z, A-X, A-Y, A-Z, specifying the translation and rotation in the axis-angle format relative to the global coordinate system [CMU09]. The format also provides the possibility of collecting other and differently formulated data – m.in:

* rotations and translations according to the local coordinate system,
* residual errors,
* forces and moments relative to individual local or global axes,
* scaled and unscaled samples from analog channels,
* a set of cameras recording a given frame,
* marker positions according to the local or global coordinate system,
* Pressure point positions according to the local or global coordinate system.

In the static data section of files, it is possible to define new, user-specific headers – their unified format allows the reading software to skip unfamiliar headers. On the other hand, the dynamic data section covers only the above-mentioned types of information, collectively referred to as DOF ("degrees of freedom").

## Publicly available datasets

Motion datasets made available to the public are only to a limited extent able to help determine the appropriate shape of the database organization solution for a generic motion analysis and synthesis system. In most cases, these files are in the nature of a set of files (binary or character). Moreover, portability and interoperability do not seem to be a priority for these datasets, so often, despite the fact that the datasets are based on standard file formats, detailed information about the sizes or conventions adopted is not readily available.

### HumanEva I

Video data acquired in uncompressed form and then compressed using the XviD codec. [HuEv09] [SiBl06]

General naming pattern: RodzajAkcji\_NrPróby[\_NrKamery] - the latter, in relation to video data.

The sets are grouped at the upper level among the individual actors. For each of them, a directory structure has been created, distinguishing the type and use of the relevant records:

* video recording,
* motion recording records (C3D files) from Vicon,
* calibration data of video cameras (. CAL),
* synchronization data between the video image and the motion recording system (. OFS),
* A file with the .mp extension (from *Model Parameters*), containing a set of 28 anthropometric data of a given performer.

The set of information from individual trials is different, which is due to their different applications:

* Sample 1: data for training and verification (division into two equal appropriate parts) - movement record and video recording,
* Sample 2: data for testing motion recognition algorithms from images - only video recording is available,
* Sample 3: data intended for motion analysis - only the motion recording was recorded

In addition, there are 3 sets of background images (from the beginning, end, and middle of the session).

The XML-based format used to evaluate the recognized movement describes frames, each of which captures a pose by determining the three-dimensional location in space of each of the 20 joints.

### HumanEva II

On the other hand, the HumanEva II [HuEv09] dataset contains only color video, uses a larger number of motion capture cameras (8 instead of 6) and uses hardware synchronization.

The units used in calibration are millimeters, and the error for a 2D image is expressed in pixels.

The data were made available in C3D, MAT formats, and supplemented with a VSK definition specifying the kinematic model and marker positions. In addition, as in the HumanEva I collection, there is a model parameters file for each performer, which includes a set of parameters that is about twice as rich.

The images that make up the video sequences are physically saved as sequential frames in png format.

### Gatech HID

Human Identification at a Distance is a project on the identification of people at a distance, based on gait analysis - in particular, based on static body parameters and stride parameters [GaTe01].

The database contains video recordings and motion capture. Data from 20 subjects were collected, the list of which is the root of the structure of the collection. The data consists of:

* video recordings from 3 different cameras, in .avi format (DV codec)
* motion records in Maya format (Maya\_data - binary .mb format and Text\_data - the latter in .mot (LightWave motion file?) text format that allows playback using a shared program).

Motion capture data is given in meters, and the coordinates of the scene space are given in inches.

The text format shows successive paths (with a length consistent with the number of frames declared), where each path can specify either translation (in the form of a vector in space) or rotation (in the form of 4 floating-point numbers).

The format does not introduce any skeletal model in the sense of a hierarchy of named elements, much less uses ancestor-relative coordinate systems. This can be seen by observing that text-based models are resistant to changes in path names or their order from one to another. On the other hand, the translation path and the rotation path for the individual elements must follow each other directly.

Translation numbers in metres, with a precision of 6 decimal places.

### HDM5

Free motion caputre database for research purposes [MRCE05]. A Vicon system was used to capture the 3D movement using 40-50 markers and 12 cameras at 120 Hz.

The data is available in C3D and ASF/AMC formats. Thus, the dataset offers both "raw" information - derived from markers, and processed information - in a form based on a skeletal model. ASF/AMC introduces an explicit skeletal structure, while in C3D hierarchy information can only be indirectly determined based on marker names.

Lengths are given in centimeters and angles are given in degrees. In the model, 31 joints and bones were distinguished. A joint is assigned to the bone to which it is external (distal).

### KUG

The KUG motion dataset includes video and mocap data for 14 types of normal movement (gestures, movements) performed by 20 older people (aged 60-80 years) and 10 types of abnormal movement (accident-related - falls) performed by 10 younger people.

The 3D movement was recorded using a system of 33 markers and 12 or 8 cameras operating at 60 Hz, while the 2D video image was recorded using a stereoscopic array of 3 cameras.

Filenames contain attributes such as: object index, session number, gender, age, "gesture" number (movement made), camera direction, frame number, file type.

### CASIA Gait

CASIA Gait is the closest topic to the project among the family of biometric datasets made available by CASIA (Chinese Academy of Sciences) as part of biometrics and security research [CBSR05]. It currently consists of 3 data sets, recorded in 2001 and 2005. Datasets A and B contain video from different perspectives, while dataset C contains images from infrared cameras. The data was made available in the form of files arranged in a directory structure - according to the performer, the activity and the perspective. Dataset A was saved as a series of PNG files corresponding to successive frames, while Datasets B and C were saved as .avi files. Silhouettes extracted from the image of the figures were also made available in the form of separate files.

### CMU MoBo

The database stores only images recorded for different performers and different types of stride (in treadmill movement) [GrSh01]. The images were also made available in the form of extracted silhouettes. There is no other measurement data. No database management system. A simple structure of a data set, based on a hierarchy: object – activity – view – image. As it seems, this dataset is not publicly available at the moment.

### CMU-MMAC

The database contains a record of the behaviour of people working in the kitchen scenery and preparing various types of food [THBM08]. A dataset consists of files separated into appropriate folders. The following were registered:

* image (3 cameras with high spatial resolution (1024 x 768) and low temporal resolution (30 Hz), 2 cameras with low spatial resolution (640 x 480) and high temporal resolution (60 Hz) and one transmitted performer camera - with low spatial resolution (640 x 480) and time resolution (12 Hz).
* Audio - 5 stationary microphones and one performer microphone
* motion capture - a set of 12 Vicon cameras with a resolution of 4 Mp and 120 Hz.
* 5 three-dimensional accelerometers and gyroscopes.

The computers used for registration were synchronised using NTP. It turned out to be necessary due to the very large data stream generated by all the above-mentioned cameras (over 100 MB/s).

Motion capture data is stored in C3D format, Vicon's V-format (wireframe-referenced data) and in a format specific to the Vicon iQ software.

The movement of the figures was recorded with a set of 40-60 markers and the interpretation of the movement as the movement of rigid solids with 15 to 22 segments.

# Context – list of modules

The basic division into architectural components distinguishes the Database System - SBD (offering interfaces to data access and a user interface for searching the data set) and System Services (US) referring to it. One of the UCs will be the Pre-Processing Module, through which the underlying data will be transferred to the SBD. This is a very specific element and due to its low stability (frequent format updates), it has been separated in the project plan from the SBD, assuming consistency with the Database System at the level of the general interface. The figure below illustrates the location of the SBD component against the background of other areas of research work, resulting in appropriate System Services. For the sake of clarity, the schedule periods of their implementation expressed in months from the start of the project are given for individual areas. Substantive and schedule dependencies result in possible flows of information between areas, where the official availability of the result of a given work can be distinguished (continuous arrow) and informal information from the partial implementation of parallel works (dashed arrow).



Drawing 1: Substantive links of the designed component with other areas of the project and the software developed within them

# Component Functionality Specification

In this chapter, the functionality of the "database" component will be presented. Therefore, it should be emphasized that the purpose of the chapter is not to specify the requirements for the entire designed system. It should be noted that the structure of the system outlined in the project assumptions provides that the database component will also include a user interface to the part of the system's functionality that depends directly on the data collected here. Hence, we will also be dealing with actors who are users of the system - and not only with the rest of the system components.

It is worth noting that the database design is drawn very early in the project life cycle, well before the architectural design of the entire system is completed. Hence, for the purposes of designing a database component, it will be necessary to give a rough indication of the functionality of the entire system. These issues will also be covered here, but only to the extent required by the design of the "database" component.



Drawing 2: Transparency of the data origin for system services consuming motion data

It is important to note that not every configuration variant of the software created as part of the project will be forced to use a database. Some variants will probably be satisfied with processing a small number of instances stored on the local disk, i.e. they will not refer to the database. From the point of view of the system service (with the exception of possible services dedicated to searching or feeding the database), it will be possible to talk about the transparency of the source of data origin (see Fig. 2), i.e. a significant part of the system's functionality will not have to be aware whether a database component was involved in obtaining the data.

Due to the above assumption, a significant part of the data collected in the database will remain in the form determined by the file formats – own or standard. This means relying on *Binary Large OBject (*BLOB) and/or *Character Large OBject (*CLOB); generally speaking – *Large OBject –*LOB when defining data storage structures in the database. In addition, data that is particularly important for search and classification will be duplicated ("promoted") by placing them in dedicated tables and columns in the database. The latter form of data can be used efficiently and effectively when processing database queries. On the other hand, the content of the LOB will be opaque for the database component – at most, information about the items inside the LOB will be stored, allowing the system to retrieve only the indicated fragment of them. The actual processing of LOB data will be carried out by the appropriate System Services, implemented based on a high-level programming language.



Drawing 3: Architectural Dependencies in the Vicinity of the Motion Database component

Therefore, if there is a need to process large amounts of information from LOB elements for the purpose of selecting and downloading motion samples from the database that meet the relevant criteria, it will be necessary to use a search system service, co-located with the database (here called the analytical search service). The interface should be designed to be open, i.e. so that further services can be defined to meet the new search criteria. Such a set of search services would determine, in addition to the saved data access interface, the data source for the user interface for motion data mining. Appropriate queries would be directed to the server to both of the above-mentioned types of interfaces – the data exchanged between them would be the master keys of data elements that meet specific criteria.



Drawing 4: Initial Summary of Use Cases - Diagram

It is assumed that from the point of view of the user interface, the properties calculated by the services executing analytical queries will be represented in a uniform way with the data directly collected in the database and will allow for the definition of criteria based on both of these types of data sources.

The database system will be subordinated to three areas of functionality, referred to as System Services:

* Medicine - focused on the analysis of the functioning of the musculoskeletal system from the point of view of orthopedics.
* Entertainment - processing of motion data for the purpose of creating animations (games, advertisements).
* Scientific Research - psychology (behaviorism, virtual reality), cognitive sciences, movement analysis for computer vision.

The order of the above-mentioned System Services corresponds to the priorities of the prototype designed here. In addition, interfaces for future applications not covered by the System Services of the current prototype will be considered. On the other hand, an indispensable element of the prototype, independent of the functionality built directly on it, will be the public Multimedia Motion Database.

In general, the scope of the designed database component can be characterized as follows.

* Recording of raw data from the recording equipment (after they have been processed by the Pre-Processing Module). Underlying data can serve a variety of purposes; For this reason, no attempt is made in this representation to arrange the data in fine-grained structures subordinated to any classification criterion. This data will therefore be streamed directly (in the form of Binary Large Objects (BLOBs) or Character Large Objects (CLOBs)). Individual recorded streams will be provided with synchronization data. The assumption is to maintain full information on the synchronization of individual observations of the subject's activity (as well as possibly registered external stimuli affecting it). The data obtained from such observations will be referred to in the following part of the document as Dynamic Generic Data (DDG), to emphasize their independence from the area of application.
* A record of Static Generic Data (SDG), characterizing the entity whose movement we are recording and the parameters of the observation itself.
* A record of interpreted data created by processing the underlying data by the appropriate tax office.
* A user interface for searching and retrieving underlying and interpreted motion records. The use of technology to provide online access is envisaged here.
* A user interface for editing data specific to the application areas to be implemented.

In addition to the above, the procedures for creating backup copies, database restoration and security policy will be defined.

From the point of view of the Database System, a number of components will be characterized by a uniform nature of interaction, including input and output through programming interfaces, the exact shape of which will be determined by the schematic design. However, each of the above services requires further refinement of data manipulation requirements so that the database structure offers optimal access performance and ease of processing.

Taking a more detailed point of view – the following modules and sub-modules belonging to the database component or constituting its immediate surroundings can be indicated:

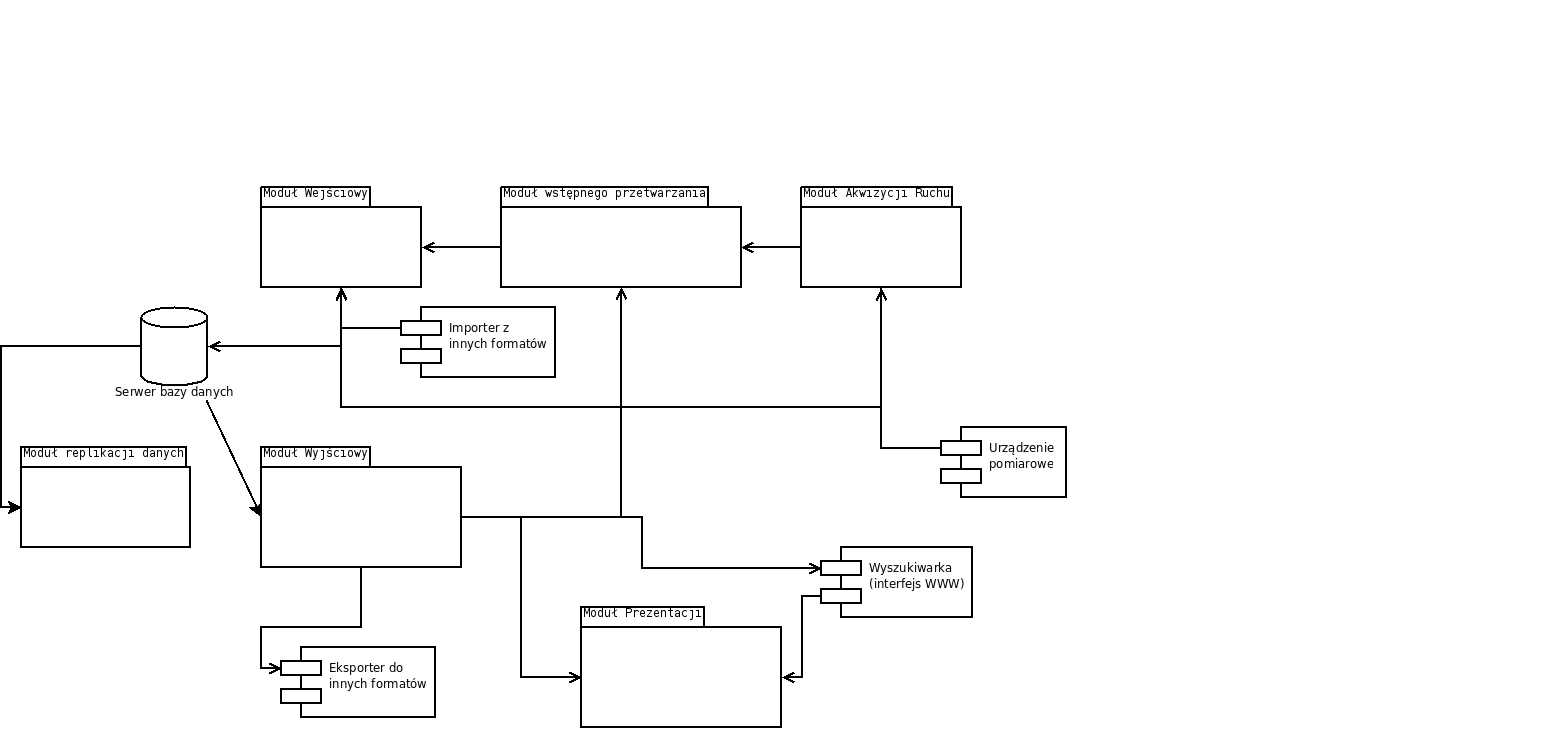
* Database System

1. Data Input Module
2. Components of Generic Dynamic Data Recording
3. Static Data Storage Component
4. Components of Processed Data Storage
5. Data Output Module
6. Data Import Module
7. Data Export Module
8. Data Replication Module

* Data Acquisition Module
* Pre-Processing Module
* Presentation Module

1. Search Component

* Data Replication Module



Drawing 5: SBD Architectural Environment and Data Flows

It can be assumed that all SBD modules are located on one machine – the server (as emphasized above, this also applies to the analytical search service). Other modules can connect to it and, after successful authentication, obtain the data stored in the database. The communication connections between the modules are shown in the diagram.

The Input Module is prepared to receive the recorded data either via the Pre-Processing Module or directly. At the same time, there may be various System Services that will process data and create new data. Such services can also expect the SBD to be prepared to record the results of their work.

## External actors

In this section, we will introduce the entities outside the boundaries of the Motion Database component that will interact with it. Their interfaces and/or communication requirements will be characterized.

### Users

The following users (human foreground actors) were identified at the initial analysis stage.

**Online user** – a user connecting to a reference motion database maintained by PJWSTK and collecting motion data obtained in its own acquisition laboratory.

**Acquisition Laboratory Operator** – a user who enters the recorded motion data into the database.

**Physician** – a priority (from the point of view of the area of medical applications) representative of the domain users of the system. It complements motion acquisition data with domain data (m.in. diagnosis). Searches the case database for specific types of domain data (e.g., diagnoses, anthropometric data) and medically relevant motion data.

### Cooperating Components

The feasibility study document [PJWS08] indicates the following system components requiring interoperability with the Motion Database component, which can be referred to as System Services (US).

**US - Pre-Processing Module** – Realizes low-level processing (shaking (jitter), cleaning, normalization, representation). It is a direct recipient of data from acquisition devices. After processing, it transmits the data to the Database component via the appropriate programming interfaces.

**US - Data Segmentation and Classification Module** – Performs additional data interpretation tasks that are not included in the pre-processing phase.

**US - Analytical Module** – Performs the analysis of kinematics and dynamics of movement, including the detection of neurological and motor abnormalities, comparison of movement patterns of people, especially sick and healthy people, determination of new symptoms of abnormal movements.

**US - Motion Synthesis Module** – Based on the definition of the character model and the movement segments extracted from the database, generates a definition of the character's movement.

**US - Data Interpretation Module** – Processes raw tag data with information about their location, or skeletal data in the form obtained from the Acquisition Point software and data on muscle activity, to a form interpreted from the point of view of the Skeletal and Muscular Model.

**US – Analytical Search** – a service related to the Analytical Module, which is an element of an extensible search mechanism offering specific data search criteria based on values calculated from LOB data blocks.

**US – Exporter to Other Formats** – Reads the data stored in the SBD and converts it to the form expected by the consumer.

## Use Case Specification

The current baseline list of use cases includes the following.

**Recording of motion data**

Actor – System Service

**Reading motion data**

Actor – System Service

**Motion data mining**

Actor – Online User

**User authentication**

Actor – Online User

**Query definition**

Actor – Online User

**Entering acquisition parameters**

Actor – Acquisition Lab Operator

**Performer Data Input**

Actor – Acquisition Lab Operator

**Editing Attributes**

Actor – User of domain data

The relationship between these functional units is illustrated in the diagram below.



Drawing 6: Preliminary Use Case Model Diagram

Within the use case specification, the use case scenarios outlined as indicated in the diagram above, structured in the form of specification templates that are an adaptation of the forms proposed in the paper by Alistair Cockburn [Cock04], presented in the table below.

|  |  |
| --- | --- |
| **Name** |  |
| **Actor/Actors** |  |
| **Included Cases** | <Skip if not applicable> |
| **Extended cases** | <Skip if not applicable> |
| **Case - generalization** | <Skip if not applicable> |
| **Prerequisites** |  |
| **Post-Completion Conditions** |  |
| **Base scenario** |  |
|  | |
| **Variant scenarios** | <Skip if not applicable> |
|  | |
| **Specific requirements** |  |

## Key Non-Functional Requirements

Non-functional requirements will be specified at the detailed design stage. Currently, it is possible to indicate such requirements as ensuring security and data protection (backup copies, possible replication, introduction of a data access authorization system) regardless of the field of application.

# Database schema

The design of the database schema will be developed evolutionarily, along with the clarification of requirements and the progress of work on the design of System Services. This document contains the initial version of this project.

It is anticipated that the Database System component will need to store the following types of information:

* Baseline data from observations, including synchronized channels:
  + MoCap
  + Video
  + sEMG
  + GRF
  + Audio
* static generic data, specifying the parameters of the acquired base data (including acquisition parameters). Including.
  + Observation Laboratory
  + Date and time of registration
  + Performer data - biomedical:
    - breed
    - sex
    - age
    - Weight
    - rise
    - [diagnosis],
    - [body fat thickness] (possibly relevant in the context of sEMG studies)
  + Performer's personal data:
    - relevant personal data,
    - A copy of your consent to the study and processing
  + Configuration of recording devices.
    - for sEMG: electrode types and locations, amplifier parameters, filter settings, sampling rate.
* dictionary and grammar of motion description (if implemented)
* static generic data describing the performer in anthropometric terms
* selected types of interpreted data, which, due to the significant cost of their calculation, will be stored in the form of persistent data structures
* data structures delineating data segments (in relation to raw data as well as interpreted data)
* additional data specific to the project aspect - e.g.
  + the patient and the examination procedure,
  + Three-dimensional model meshes and texture data
* Three-dimensional model meshes and texture data
* Structures that designate data segments
* Categorization-related structures
* Structures that determine the levels of data access rights

## Initial schema design for the database

### Introductory information

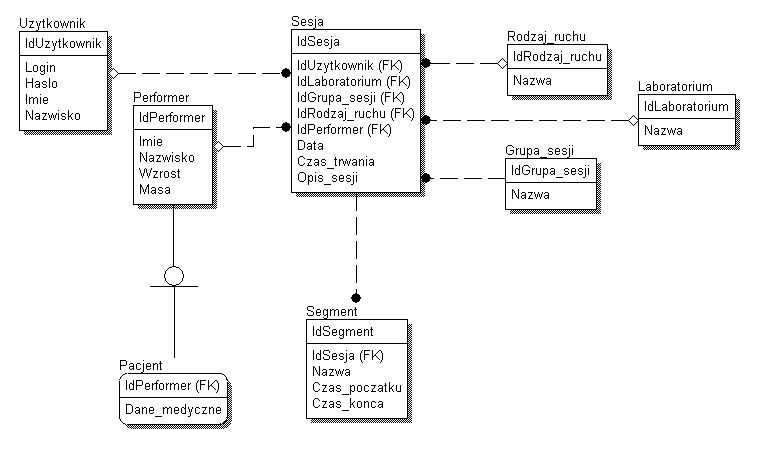
The database schema has been prepared in IDEF1X notation. Basic diagram properties:

* A rectangle denotes an entity, i.e. an entity about which we want to store information. The entities become database tables.
* Above the rectangle is the name of the entity (in the future table).
* Inside the rectangle, above the horizontal bar, there are attributes that are the primary key (uniquely identifying the entity). When designing the schema, the concept of using artificial autonumbered master keys was adopted, which will be hidden in the user interface.
* A line connecting entities indicates a relationship. Dot is the "many" side of the relationship.
* A diamond on the "one" side of the compound means that a union is not required.

For the sake of readability, for the purposes of this documentation, the diagram has been divided into several views.

### Diagram sketch for the identified elements of the problem domain

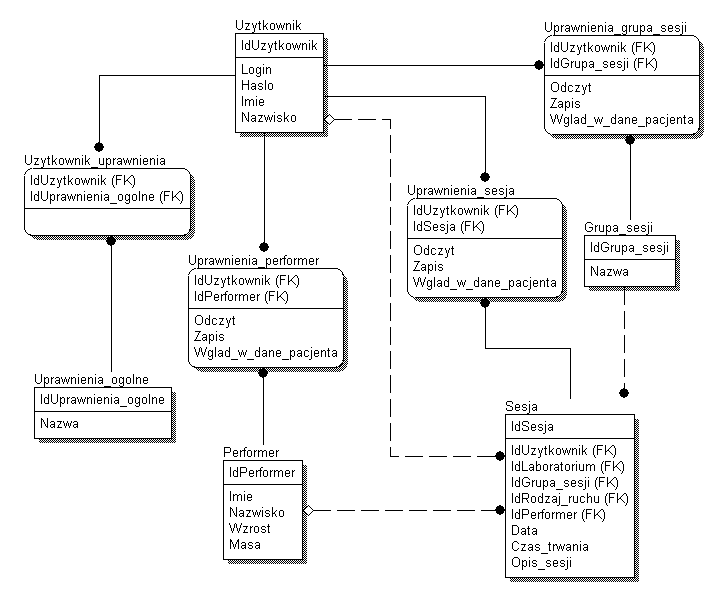
#### Session



Drawing 7: Initial Database Schema Design – Session Data

The "session" entity represents a single motion recording session. At this point, it is irrelevant which measurement techniques were used. A "performer" can be either an actor or a patient being examined. In the latter case, the data will be expanded to include medical information (individual attributes have not yet been specified). "Type of movement" is an optional dictionary that will contain records such as: walking, running, jumping. Sessions can be organized into "Session Groups", i.e. sets of measurement sessions made for specific needs. An example of such a group would be "Measurement sessions made for the needs of computer game X". A session can consist of multiple segments, i.e. separate parts with a fixed start and end time. For each session, we know the user performing the test (observer). The unit of time used in the database has not yet been determined. These can be both actual times given with a set precision, as well as frame numbers.

#### Permissions

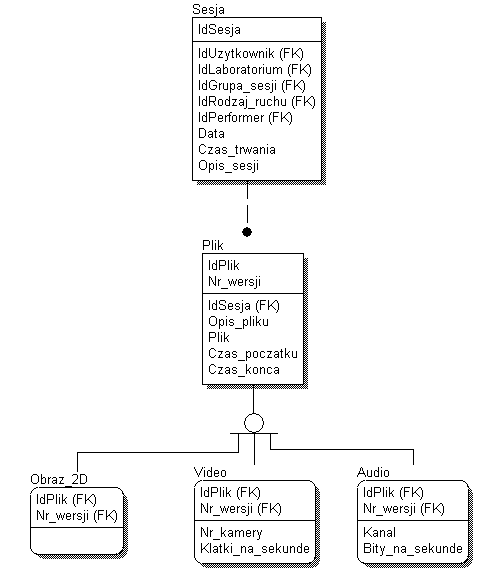
****

Drawing 8: Preliminary design of the database schema – structures defining user rights

Due to the wide variety of applications of the motion database component, it is necessary to model the authorization system as flexibly as possible. That is why it is possible to grant permissions at different levels: a single session, a group of sessions and a performer (e.g. access to a patient's data). Currently, there are 3 types of authorizations: read, write and view patient data. Other powers at this level remain an open question.

In addition, the "General Permissions" entity has been created. It reflects rights that are not tied to specific measurements or actors, and will allow you to define arbitrary rights that will then be handled by applications. Examples of such rights include "data mining" or administrative rights.

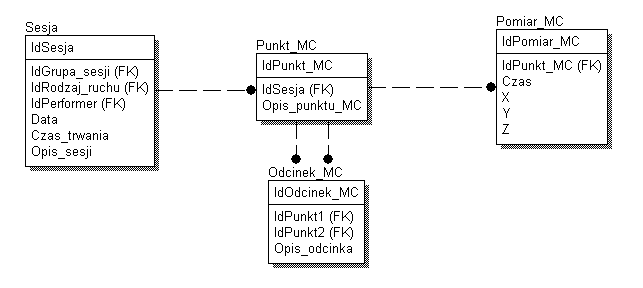
#### Files



Drawing 9: Preliminary design of the database schema – files

As a result of each measurement session, many different types of files will be stored in the database. These can be 2D images, video sequences, audio, or anything else. Depending on the type of file, its attributes vary. It is not required that the file be classified into one of the three defined categories, e.g. a C3D file will only use the "File" table. A file can cover the entire session, its segment, or any part of it. Therefore, it is important to specify the start and end times in the same units as in the "segment" table. As a result of various types of data processing operations, it may be necessary to store multiple versions of the same files. For this reason, the "nr\_wersji" attribute has been added to the primary key, which, together with the file ID, uniquely identifies the file.

#### Motion capture

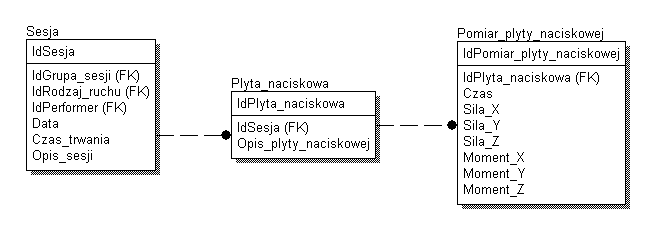


Drawing 10: Preliminary design of the database schema – 3D motion data

Motion capture measurement data can be saved both in the raw form of a file (e.g. C3D) and in the form processed into a relational database. For each measurement session, we store information about the points (markers) and changes in their location over time. Points may (or may not) be connected by segments (e.g. bones). This form of motion capture data recording will allow for easier processing and sharing, directly with the help of database engine tools, without the need to use tools that implement a specific format.

#### Pressure plates

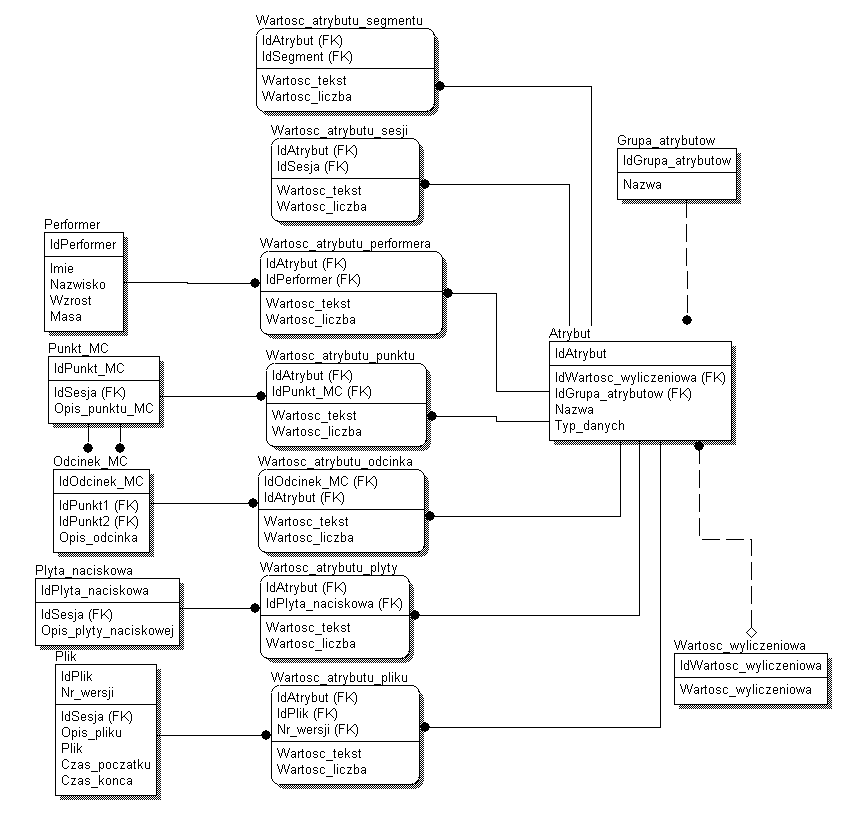
The final shape of the solution for this part of the schema is subject to similar conditions as the MC data and will probably take the same form as the database recording.



Drawing 11: Preliminary design of the database schema – data from pressure plates

This part of the diagram is responsible for the ability to write the processed data from the pressure plates to the relational database. We store information about the individual plates used in the measurement session and the change in forces and moments over time. This method of saving is an alternative to saving data in raw form (file).

#### Attributes



Drawing 12: Preliminary Database Schema Design – Generic Attributes

Because the database will be used by different users for different purposes, it is impossible to specify a set of attributes for all entities in advance. It is almost certain that as the project develops, the attributes will change. Therefore, a solution has been proposed in the data model that will allow users or designers to freely define attributes and link them to individual measurement devices, sessions, performers and data streams. Attributes can form groups, which will make them easier to classify. An attribute can be set up with a closed set of enumeration values defined by the administrator, which will ensure, where possible, a uniform way of specifying them by annotating users, who will select the value from the list instead of typing it.

## Assumptions for detailed design

A significant challenge for the design of the data schema will be the genetics of the system being built and the data stored in it, while maintaining the efficiency of access for the basic applications carried out as part of the current project. An important assumption resulting from this is the lossless preservation of the complete set of data obtained from the observations – probably in a form similar to the one in which they were recorded (with the accuracy of their possible purification at the stage of initial processing). This entails the need to record a significant amount of data for each observation – probably in binary form, with high temporal resolution. Thus, mapping individual frames as separate tuples in a relational table for the entirety of the collected data would entail too high an overhead. Hence the concept of quite intensive use of BLOB data types (large binary objects), which allow to store in a single field (from the point of view of the database management system) a large number of consecutive indications of the measuring device (perhaps even covering the entire length of the observation session). As mentioned above, such a file may cover the entire session, its segment, or any part of it.

A similar concept was also used in [EMSS05], where a relational database was used, in which short time (approx. 0.5 s) but multi-frame motion segments were stored in the form of single BLOB values. On the other hand, dedicated database columns with traditional data types define important parameters (in this case, positions, rotations and speeds) that identify the initial and final state of a given part of the movement. In the case of our system, the situation is partly complicated by the fact that the data can be used for a variety of uses, which may involve different segmentation of motion from a single session. This may force a situation in which the data stored in other columns of the database not only annotate the BLOB data with information that facilitates their selection (as a whole), but even identifies specific sub-sequences constituting a selected part of it.

The above-quoted publication also raises the issue of the complexity of criteria in queries, e.g. when determining similar fragments of movement for the purposes of its synthesis. This problem may give rise to the need to support search mechanisms with definitions of perspectives or functions that will make data available in a form that simplifies the formulation of search conditions on them.

Another publication [GPSB02] highlights the need for an open and flexible system for annotating and classifying motion data. It indicates the following requirements:

1. Ability to support several motion data classification schemes.
2. Extensibility with new classification schemes.
3. Provide vocabulary for each of the classification schemes.
4. Allowing the collection of data characterized only in certain respects.

The implementation of this postulate is illustrated in the "Attributes" section of the scheme. In addition to containers for attribute values, tables store metadata, i.e., names, types, and groups of attributes. Adding a new description criterion then adds a new value to the metadata set and does not require modification of the schema. Attribute groups can be associated with motion segment classification categories and/or types of recorded parameters, which will provide better uniformity in terms of the set of attributes that describe each instance.

# Database Component Interfaces

## Visual Interface Concepts

The user interface for motion database exploration should offer the possibility of intuitive selection of material based on numerous criteria, based on the stored or calculated properties of the recorded motion. It seems useful to define the selection criteria freely, as well as to select specific data in order to download them and/or treat them as a model for further searches. In this case, we are talking about intentional and extensional navigation, respectively. Due to the characteristics of motion data and typical search scenarios, most often the search condition will be not so much to find records identical in terms of a given feature, but close to the searched value. Hence, there is also a need to be able to present the results in a way that would illustrate the degree of similarity of individual data instances with the searched pattern. A useful model of such functionality can be a generic data mining model designed for the Mavigator [TrSu04] system. It derives from Charles W. Bachman's concept of navigation-based data processing in a network data model. This approach, which has its roots in the networked database model, seems very promising today, given the success of the concept of the World Wide Web and visual interfaces to data. This approach is able to combine the mechanisms of a query language (determining, based on the definition of the data schema, selection criteria) and an interface for data exploration (browsing) – both based on a user-friendly visual interface.

The interface concept distinguishes between the so-called intensional and extensional navigation. Intensional navigation allows you to visualize the database schema in the form of a graph, and to impose constraints on the schema in accordance with this schema, and navigate by observing the number of instances selected by the criteria for each entity. A prototype implementation of the interface in the Mavigator tool is shown in Figure 4.



Drawing 13: Intensional navigation and cart views in Mavigator. Image Source – Figures 1 & 4 from [TrSu04]

An alternative to applying conditions through dialog boxes is to place them directly on the diagram – as it was implemented in the "Visual Expression Builder" component designed for code specification in the UML environment [HCS08].

Extensional navigation, on the other hand, allows you to navigate through a graph illustrating the state of individual objects. It is useful when it is not possible to determine in advance the selection criterion in terms of the value of the relevant attributes and the presence of the appropriate connections.

Another potentially interesting extension of the visual interface designed by the Mavigator project that seems adequate for comparing numerical quantities is the visualization of selected instances on a two-dimensional graph. It allows for a graphical presentation of the convergence of individual instances with the indicated pattern based on two different numerical criteria at the same time.

In addition to intent and extensional navigation, the Mavigator interface introduces the concept of a "shopping cart", in which a given user can remember selected data for later exploration. The approach to interface construction can be described here as schema-driven, as the basic form of a search interface is based on a graph, the structure of which is implied by the schema of the searched data. Namely, vertices and arcs of a graph used to search the database can respectively illustrate classes and associations or (also equipped with typological information from the schematic) objects and connections of specific instances. The basket is useful both for storing intermediate results during a single data exploration session, but also for dividing the work between stages carried out in different sessions and for defining your own handy data set. It allows you to create an ephemeral or permanent set of references to selected data. Corresponding objects can be placed in the basket either individually, in the case of extensional navigation, or by indicating a set of objects that meet the criterion. In both cases, it is worth considering support for the opposite action, i.e. dragging the resource from the cart to the exploration window.

On the other hand, taking into account the fact that, in contrast to the above-mentioned approaches, we are dealing here with an established problem domain and that the facilitations related to the data access layer, query optimization and management of intermediate results may prove to be crucial, it should be taken into account that significant simplifications are necessary in relation to the above-mentioned concepts, bringing the user interface closer to a form-based solution.

Given the complexity of the schema for a fully-fledged repository of character motion data, it is to be expected that from a user-friendliness point of view, a selective approach to the metadata that can be searched and navigated will be essential. It will be necessary to determine whether this type of selection can be made in advance for all users, or whether it should be configurable for individual users. These observations suggest that it may be practical to define virtual perspectives (views) that have been entered into the schema specifically for the search interface [TrSu06].

## Application Interfaces

The key to the development of the system will be to define a programming interface for BDR resources, which will enable the integration of other tools (e.g. a motion editor) and facilitate further evolution and diversification of the online database search client.

It is assumed that the database resources will be made available in the form of coarse-grained services, in accordance with the Service Oriented Architecture approach, although without deciding on the specific choice of technology at this stage. It must reconcile the requirements of flexibility and platform independence with operational efficiency, taking into account the transmission of large volume binary and character data.

# Technical assumptions

## Volume of data considered

Publicly available examples of datasets with motions (e.g., Brown, CMU) and are the first element that provides an initial general look at the specifics of using this type of dataset. Using the CMU database as an example, you can specify the following data volumes for a single observation. Each sequence has an average of 6-7 representations in the following formats:

* asf - skeletal definition, 30 bones, created directly in Vicon BodyBuilder one file for one character;
* amc - motion data each has an average of ~3000 frames, which gives ~ over 2MB/motion sequence;
* tvd - raw video data for 3D reconstruction from the Vicon system ~12MB/sequence;
* c3d - processed data in the form of 3D trajectories usually serve as input for kinematic data analysis ~2MB/sequence;
* mpg - video recording of data for demonstration purposes in the studio ~5MB/one visualization;
* avi - visualization of MoCap data for demonstration purposes in a synthetic way ~3MB/sequence.

In our system, we anticipate the collection of the following types of data:

1. MoCap (e.g. in c3d format)

B) Detailed data streams from video recordings in the form of mpg files + description of the conditions in which the measurement was made (status, sequence number, angle of observation, optical parameters, etc.) + video streams of silhouettes generated from each video sequence. It can be assumed that the video recording has 4 cameras, which gives 8 video streams + streams with synchronization tags with MoCap data.

C) Multi-channel audio, let's assume a maximum of 32 channels of sound synchronized with MoCap data (can be a verbal commentary or background music, or communication between the actor and the session director).

D) Surface EMG (sEMG) to measure neuromuscular activity.

Assuming a maximum of 32 channels of 1kHz/channel/resolution of 16 bits, we get 32 analog interference streams from overlapping action potentials of motor units (raw EMG data) + possibly 32 rectified signals + possibly additional interpretation in the form of purified data (RMS) and normalized to the reference amplitude value of the EMG signal (%MVC / %RVC).

This results in a single measurement of up to 32 analog signals in 4 formats (raw data, rectified data, RMS data, normalized data).

E) GRF data from a dynamometer platform consisting of two Kistler plates.

Assuming that one board is equipped with 4 piezoelectric transducers and we have 6 channels per board with force data: Fx, Fy, Fz and moments Mx, My, Mz at an average acquisition speed of 200 pulses/second. This gives us 12 streams of raw data information.

F) When measurements of the action potential of the EMG signal and the reaction forces of the ground are to be synchronized, a stream of synchronizing data is added.

G) Acquisition parameters for each type of data corresponding to the type of data to which the parameters relate (e.g. parameters of cameras for video recording, microphone, etc.).

H) In addition, the possibility of recording: EEG, MRI, PET (channels that are used with EMG can also be used for EEG in the simplest version)

On the basis of the attached sketch of the underlying data sources, it is possible to make preliminary conclusions as to the type of base data structures that will be processed in the system.

To sum up, the volume of data for several thousand sessions should be counted in tens or hundreds of gigabytes.

## Choosing a Database Management System

After analysing the database management systems available on the market in terms of their use in the project, we suggest using Microsoft SQL Server 2008 software. The most important arguments in favor of this choice:

* High performance compared to open source products. Although free products have an advantage in some simple database performance tests, commercial servers are more efficient when it comes to complex operations with large data sets.
* Moderately priced compared to other commercial database servers (e.g. Oracle).
* Can be integrated with all popular development environments.
* SQL Server is very good at storing and sharing large data objects (LOBs). In the 2008 version, an additional option, the so-called Filestream, was introduced, which further improves the efficiency and convenience of storing large data objects. Filestream combines the convenience of storing large data objects in a database with the performance only achievable when storing files directly on disk. The combination of these advantages is made possible by the fact that the data is physically stored in the file system, but can be accessed from a database server, similar to traditional LOB data. The size of the stored files is limited only by the available space on the server's disks. Any administrative operations (e.g., backup) involve Filestream data. Operations on this type of data (e.g. searching, indexing, inserting, modifying) are possible both with the database layer software and in the higher layers of the application.
* Very good scalability of the system.
* Transact-SQL as a good, efficient, and well-documented language that is a procedural extension of SQL. Since many of the system's use cases may prioritize data processing performance, it will undoubtedly be necessary to perform at least some of the processing on the database server side.
* Support for data replication, both unidirectional and multidirectional, based on the most popular models.
* Very good documentation and technical support.

## Server Hardware Requirements

Due to the large amount of data stored, as well as the need to ensure high performance of reading and writing large data sets, hard drives will be a key element in choosing the right server for the system database. Taking into account the size of the motion databases we analyze, ensuring data security through redundancy, and system overhead, we suggest a total disk capacity of 2 TB. It is advisable to use several (6-8) disks to allow the creation of arrays. It is important to use the fastest possible drives.

Taking into account the small number of simultaneous user connections, the number and speed of processors will be less important. Sample configuration: 2 \* Intel Xeon 2.5 GHz QuadCore. The recommended amount of RAM according to current assumptions is 16 GB.

## Operating instructions

Given the large amount of data, it is imperative to provide a suitable device to back up the database. Such a device (streamer) should offer capacities of several hundred gigabytes on a single medium. The speed of the backup device will not be a key factor. Once you have determined how often and how each user uses the database, you will need to propose an appropriate strategy for the placement of files on the server's disks. By separating the data on individual aspects of the project into separate files, it will be possible to manage them independently, e.g. make separate backups. This configuration, along with the use of differential backups, will significantly reduce the costs associated with performing and storing backups.

## Plugin-based architecture postulate

Integration of System Services can be organized in the form of plugins for SBD. Thanks to it, in the future, it will be possible to create any system service that will be able to use data in the SBD as well as store new data in it. Plugins will also be able to use each other's services. In this way, the system will be ready for expansion, flexible and as easy to modify as possible. The usefulness of this type of solution has been commercially proven in many business applications, such as the well-known Eclipse programming environment. The design of the plugin system and their architecture will be the subject of a separate study.

# Summary

A natural starting point for the detailed design of the database schema will be the form of the input data from the observations, because it will determine the exact information scope of the collected data. The data formats presented represent different approaches to how the measurement data is arranged. There are options such as saving a series of readings of individual tags in sequence, or grouping readings of all tags into frames. Finally, following the example of the C3D format, you can consider placing a set of data from various instruments inside the frames.

Also, the structures responsible for classification and annotation can only be specified on the basis of more precise requirements on the functionality of the system.

Architectural decisions regarding the priority configuration variants of the tool and ways to achieve its extensibility will be important for determining the structure of the database component. M.in. It is necessary to decide whether a plugin-based approach will be used.

# Glossary of terms

**Basic Data (or Basic Measurement Data)** - data from acquisition, i.e. data with information content corresponding to that obtained from recording devices. The result of the cleansing or translation of the data format carried out by the Pre-Processing Module will continue to be classified as Base Data.

**Domain Data** - data that is an extension of Generic Data and is related to those that are relevant only to a specific field or fields of application (e.g. medicine, creating three-dimensional animations). An example of Domain Data would be a patient's medical history.

**Generic Data (DG)** - data on the movement, the character performing it, or the observation session, characterized by neutrality towards the area of application of the recorded motion.

**Interpreted Data** - data which, based on the created models and the collected static data, were created in order to obtain additional information not available directly in the acquisition data (base data). E.g. DI They can determine the movement explicitly in terms of the movement of individual joints and the actions of individual muscles, based on the skeletal and muscular model and the anthropometric parameters associated with it.

**Dynamic Generic Data (DDG)** - data from registration (base or interpreted), neutral to the field of application, describing the behaviour of the tested character (and possibly also significant changes in the environment in which it moves), characterized by variability over time during the observation session. Examples of DDGs are the underlying and interpreted character movement data, video, audio recording, sEMG and pressure plate read sequences. E.g. skeletal form of movement data, *rectified EMG data*, silhouettes from video recordings, mixed data, e.g. time dependence of muscle contractions on the moment of gait.

**Ground Reaction Force (GRF)** – a motion acquisition device that measures the force of pressure on the ground.

**Pre-Processing Module (MWP)** - is one of the System Services, but specific in terms of its role in the life cycle of measurement data. MWP processes data from the Acquisition Point before they are initially entered into the Database System.

**Performer** - a person whose movements are recorded in the process of motion acquisition.

**Acquisition Point (PA)** - a set of measuring laboratory devices, capable of carrying out measurements and pre-processing of assumed quantities and making them available in a standard data recording format. In the next step, the data from the Acquisition Point will be processed by the Pre-Processing Module (MWP).

**Data segmentation** - dividing data into homogeneous groups according to a specific criterion. We distinguish, m.in, **segmentation by measurement type**, in which we divide data into groups concerning individual measured quantities or types of measurement. On the other hand, an example of segmentation according to other criteria would be isolating, in a sequence of frames representing the character's behavior over time, a subsequence that constitutes a homogeneous (from the point of view of a given segmentation criterion) fragment of the character's movement. Due to the potentially larger number of criteria, it should be assumed that a given piece of motion may belong to more than one segment, and that such segments do not have to be in a whole-to-part relationship with each other. Depending on the segmentation criterion, there may also be a situation in which the division of data will not be a sub-sequence of dynamic data, and the method of separating groups will be more sophisticated.

**sEMG** (surface electromyography) – a technique for recording muscle activity based on sensors placed on the surface of the body.

**Motion Recording Session** - an uninterrupted recording of a character's behavior (with a duration of a few seconds to several minutes), thus characterized by a single set of Static Generic Data.

**Static Generic Data (SDG)** - Generic Data describing the characteristics of the performer and the configuration of the recording devices, common and unchangeable for the entire observation session. E.g. performer identification data, descriptive data of the session and MC performer, anthropometric reference data (joint range of motion, DOF number, bone length.

**Database System (SBD)** - an architectural component of the Human Figure Motion Analysis and Interactive Synthesis System, based on a database management system, offering the functionality of permanent collection, retrieval and retrieval of motion data. The design of the database schema and interfaces will be subordinated to the requirements of the System Services constructed within the system and the anticipated prospective applications of the tool. Due to the way the tasks are divided in the implementation of the system, a web-based user interface will also be designed and implemented in connection with the SBD component.

**System Service (US)** - an architectural component of the Human Figure Motion Analysis and Interactive Synthesis System, cooperating with SBD and implementing specific functionality of the AiSR System related to the processing of motion data.

# Origin

[BaFi08] Mark Barnes and Ellen Levy Finch. COLLADA – Digital Asset Schema Release 1.5.0 Specification. April 2008. Khronos Group Inc., Sony Computer Entertainment Inc. www.khronos.org/collada/

[C3D2009] C3D. An introduction to C3D files. http://www.c3d.org/introduction.html

[CBSR05] Center for Biometrics and Security Research. CASIA Gait Database. http://www.cbsr.ia.ac.cn/english/Gait%20Databases.asp

[CMU09] Carnegie Mellon Graphics Lab: Motion Capture and File Formats. http://mocap.cs.cmu.edu/info.php

[Cock04] Alistair Cockburn. How to write effective use cases. WNT 2004.

[CUVR09] Center for Artificial Vision Research, Korea University. Full-Body Gesture Database. http://gesturedb.korea.ac.kr/AboutDatabase.htm

[EMSS04] T. Edmunds, S. Muthukrishnan, S. Sadhukhan, S. Sueda. MoDB: Database System For Synthesizing Human Motion. Rutgers, The State University of New Jersey

[Gate01] Georgia Tech: Computational Perception Laboratory: Human Identification at a Distance - Description of Database. http://www.cc.gatech.edu/cpl/projects/hid/Description.html

[GrSh01] Ralph Gross Jianbo Shi. The CMU Motion of Body (MoBo) Database. CMU-RI-TR-01-18. June 2001. http://www.ri.cmu.edu/pub\_files/pub3/gross\_ralph\_2001\_3/gross\_ralph\_2001\_3.pdf

[HCS08] P. Habela, A. Charfi, A. Spriestersbach. Visual modelling of behaviour in data-intensive business applications. Whitepaper. http://www.vide-ist.eu

[HuEv09] Sigal, L., Balan, A., Black. M. J., HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion, International Journal of Computer Vision, 2009 http://www.cs.brown.edu/~black/Papers/EHuM\_Journal\_webversion.pdf

[MLS08] Motion Lab Systems. The C3D File Format - User Guide. http://www.c3d.org/c3ddocs.html

[MRCE05] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, Andreas Weber. Documentation Mocap Database HDM05.   
http://www.mpi-inf.mpg.de/resources/HDM05/07\_MuRoClEbKrWe\_HDM05.pdf

[SiBL06] Leonid Sigal and Michael J. Black. HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. Department of Computer Science, Brown University Providence, Rhode Island, 02912 CS-06-08 September 2006 http://www.cs.brown.edu/publications/techreports/reports/CS-06-08.html.

[THBM08] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, Pep Beltran. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database. CMU-RI-TR-08-22. April 2008. Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213. http://www.ri.cmu.edu/pub\_files/pub4/de\_la\_torre\_frade\_fernando\_2008\_1/de\_la\_torre\_frade\_fernando\_2008\_1.pdf

[TrSu04] Mariusz Trzaska, Kazimierz Subieta. The User as Navigator. Local Proceedings of ADBIS'04 Eight East-European Conference on Advances in Databases and IS, 22-25 September, 2004, Budapest, Hungary

[TrSu06] M. Trzaska, K. Subieta: Mavigator: a flexible graphical user interface to data sources. Proceedings of the working conference on Advanced visual interfaces, AVI 2006, Venezia, Italy, May 23-26, 2006. ACM Press 2006: 500-503

[Vico01] Vicon V-File Format. Vicon Motion Systems Limited Issue 2 19 September 2001. mocap.cs.cmu.edu/mocap/ViconVFileFormat.pdf

[Wisc99a] The University of Wisconsin. Acclaim ASF/AMC. http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/ASF-AMC.html

[Wisc99b] The University of Wisconsin. Biovision BVH. http://www.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html

# List of illustrations

[Figure 1: Substantive links of the designed component with other areas of the project and the software developed within them 14](#_Toc257625203)

[Figure 2: Lineage Source Transparency for System Services Consuming Motion Data 15](#_Toc257625204)

[Figure 3: Architectural Dependencies in the Environment of the Motion Database Component 16](#_Toc257625205)

[Figure 4: Initial Summary of Use Cases - Diagram 17](#_Toc257625206)

[Figure 5: SBD Architecture Environment and Data Flows 19](#_Toc257625207)

[Figure 6: Preliminary Use Case Model – Diagram 22](#_Toc257625208)

[Figure 7: Preliminary Database Schema Design – Session Data 25](#_Toc257625209)

[Figure 8: Preliminary Database Schema Design – User Rights Frameworks 26](#_Toc257625210)

[Figure 9: Preliminary Database Schema Design – Files 27](#_Toc257625211)

[Figure 10: Preliminary Database Schema Design – 3D Motion Data 27](#_Toc257625212)

[Figure 11: Preliminary Database Schema Design – Pressure Plate Data 28](#_Toc257625213)

[Figure 12: Preliminary Database Schema Design – Generic Attributes 28](#_Toc257625214)

[Figure 13: Intensional navigation and shopping cart views in Mavigator. Image Source – Figures 1 & 4 from [TrSu04] 31](#_Toc257625215)