Telemetry Data Transmission, Storage and Processing System Supporting Ski-mountaineering Race

OLDRICH HORAK
Faculty of Economics and Administration, Institute of System Engineering and Informatics
University of Pardubice
Studentska 84, 532 10 Pardubice
CZECH REPUBLIC
oldrich.horak@upce.cz

Abstract: - The article describes design of a new telemetry data transmission, storage and processing system used to transmit, store and process data measured during an overnight ski-mountaineering race. The proposed system has to solve many problems connected to the data transmission in the hard-accessible terrain, transmitted data storage and processing and some more problems originated from specific conditions of such a race. Special emphasis is given on the low-budget multi-platform solution and easily accessible technologies.

Key-Words: - Telemetry, Transmission, Storage, Processing, Network, Hard-accessible terrain, Agent, Multi-platform

1 Introduction

During the first Saturday of January many ski-mountaineers can measure themselves against the other ones in an overnight race named ”The Night of the Sealskin Tracks” since 2002. The cross-country track traverses the hard-accessible terrain of Krkonose National Park (KRNAP) located at the Czech Republic – Poland borders region (see Fig. 1).

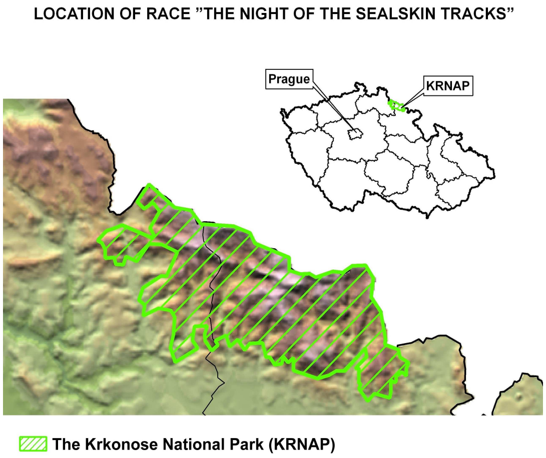

Fig. 1 - Czech Republic, location of race region

Each racer encounters many terrain difficulties and roadblocks. The race consists of several parts, each one must be passed with another style of langlauf or climbing. The track length is just 20 km and its total accumulated elevation is over 1800 m (see Fig. 2). The final score is based on the time elapsed on the track, but there can be some time-penalization for a breach of rules or for insufficient equipment (first aid kit, safety lamp, prescribed clothes…).

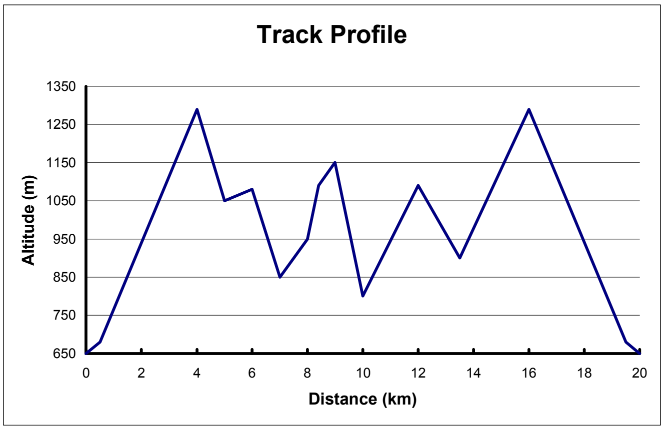

Fig. 2 - Track profile

In recent years the on-line web-based registration and the results publication is understood as an obligation. It means the system must have a public part accessible from Internet.

Personal data of racers, their start time, times at checkpoints, finish time and possible penalty points (several minutes post-added to the finish time) with a reason of the penalty must be stored. In any moment a number of racers on each part of the track must be available.

Data must be transmitted with a minimum delay. Intermediate results should be available on Internet with a minimum delay too. Because of the mountain terrain it is impossible to use temporary wired network connection, power supply is available only on a very few places along the track and there is no direct visibility between checkpoints. However, a significant part of the area is covered by GSM signal.

2 Problem Formulation
The basic problem of the solution is the hard-accessible terrain. It means temporary cable lines to connect all the checkpoints equipment cannot be laid down along the track. There is no direct visibility among places where the referees watch over racers. The only light sources available on the track are lamps on the racers' safety helmets.

So, the following questions and tasks must be solved in the above stated conditions:

- How to collect necessary data
- How to transmit collected data to the processing center
- How to process and present data
- How to provide feedback from the processing center to the checkpoints along the track.

Next requirement on proposed solution is necessity of utilization such technologies that can be easily implemented with a limited budget.

2.1 Data Acquisition
There are a several data types to collect. The first type of data records an event of racer passing the checkpoint. It is used to detect how many racers are running in the given part of the track. Second type extends the previous one - event's timestamp is added to enable calculation of checkpoint times and final time of racers. Third type of collected data is used to enter time-penalizations for breaking rules.

Other types of data keep some additional information about the race conditions (temperature, humidity…) which are not significant for the race result calculation.

2.2 Data Transmission
Data transmission is the main problem to be solved. Using some temporary wired connection is impossible in this terrain type. Checkpoints are rarely equipped with power-sources, because they are mainly placed in open terrain. The GSM signal coverage allows using cellular phone connection to transmit data, but the usage of this technology to establish connection among the entire checkpoints makes the costs very high. However, it is the only way how to pass the data into the processing center.

2.3 Data Processing and Presentation
Data processing must be done quickly. The processing center must be able to receive the data in a form of short messages with a given format. Each message would include unique message identifier, identification of source, type of the payload and the payload data. Then the data must be stored and consequently processed.

Processed data must be presented in a human-readable form on request immediately. The presentation would be reachable both for private and public use.

2.4 Feedback
Several processed data can be used as feedback information for agents and organizers along the track. The number of racers still running on the given part of track must be available at any time. It helps to have a total overview of the track and to avoid a problem of lost racers.

2.5 Limited Budget
Ski-mountaineering is an extreme sport and it is not as popular as another sports. It is one of the reasons of limited budget allocated for that race. The entire system must be able to run with minimal costs and minimal effort as possible.

Special technologies and solutions cannot be used. Proposed transmission system must be able to work on common hardware and software configuration.

3 Present System Architecture
The present system is based on a client/server architecture. The system consists of two main parts: control part and agent part (see Fig. 3). Agents are placed along the whole track. The control part consists of data layer (MySQL), executive layer (PHP) and presentation layer (Apache). This part is responsible for basic records containing data about participants of the race and incoming data from agents placed along the track. An agent has in general the following parts: data input interface, communication interface and feedback output.

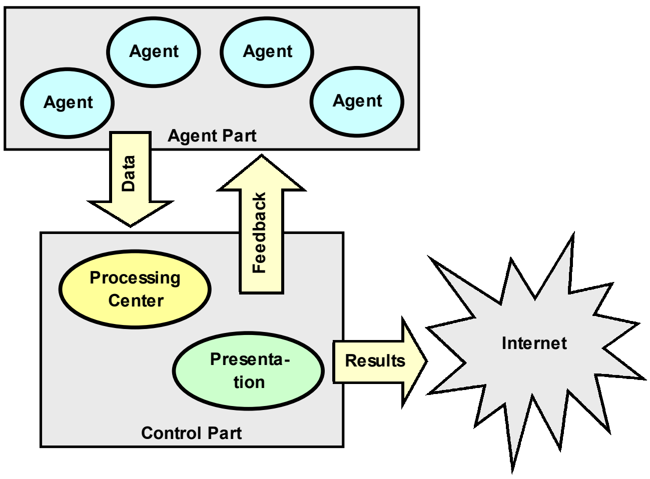

Fig. 3 - System communication

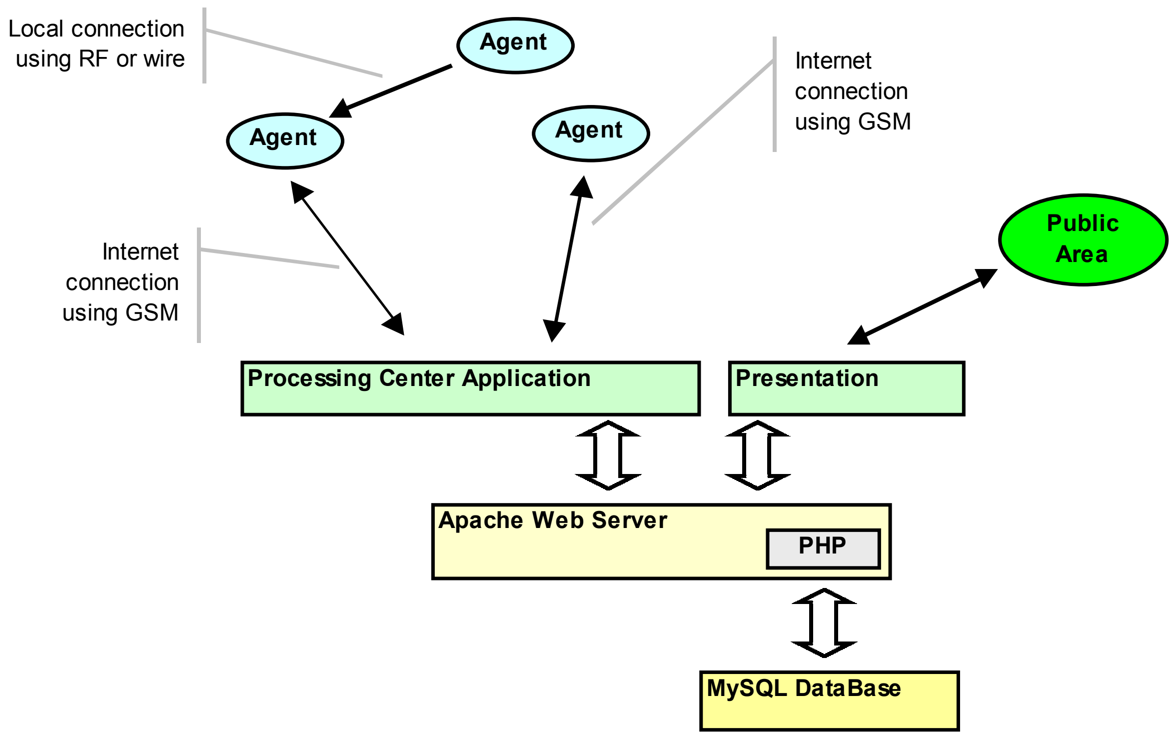

Fig. 4 - System Architecture

Several types of agents are used in according to their utilization. Measuring agent is used to record data about times of competitors on the track. Registration agent is used to keep records. Next type of agent is used for manual data entry. Agents are equipped either with one-chip sensors or with notebook with operator.

Data are transmitted from non-operated agents to operated agents by means of radio frequency (RF) transmission (see Fig. 4). This transmission is done by means of mobile Internet connection. Feedback is provided here to operators and organizers as well – all necessary data are sent back to them (e.g. how many competitors are still on the track). Agent with operator is designed as a simple web-based application. Application does not need to be installed, it can be just copied. Application uses HTTP requests to transport data from client to server [1]. This approach allows to use the application even in a local network with proxy server. When a new agent is required it is sufficient just to copy the application, enter a unique identifier (ID) of new agent and connect to Internet.

4 Design of a New System

Present solution does not take into account security of data transmission. It means it would be possible to replace original data by another data during transmission and submit false data to the control part. A possible solution is to secure communication by means of SOAP [2]. In this case advantages of HTTP communications can be kept and interface between parts of the system can be defined more precisely.

In study [3] it was proposed to use communication of the client unit and base station over SMS or short messaging service. This approach has some disadvantages. Each agent would have to have SIM card (Subscriber Identity Module), SMS is limited to 160 characters, price of sending SMS is not low, transmission speed is slower then in case of HTTP communication and mobile phone is busy until SMS is successfully sent.

Unsolved losses of network connections are the next problem of the present system although they can cause losses of transmitted data.

But system and network architecture should be designed to provide, among others, the conceptual design of the network security infrastructure, related security mechanisms, and related security policies and procedures [4].

To meet this requirement, the new solution will use SOAP and it will be based on typical architecture of today's information systems, i.e. on the n-tier client/server architecture. At least the following parts can be usually recognized [5]:

- Data layer – data management system which is able to store and provide all necessary data.
- Application layer (business logic) – processing and analytical functionality.
- Presentation layer – users interface.

4.1 Requirements on the Hardware

There is no necessity of special hardware equipment for the server computer. Server of a common configuration can be used.

However, the agent hardware must be able to establish a network connection using cellular phone or if located near by other agent, it must be able to make any connection with that agent directly and use it as a proxy server.

Server Hardware

- PC based on Pentium II or above CPU
- 512 kB of RAM
- 40 GB HDD (or RAID recommended)
- Network Interface Card(s)
- UPS
- Other common accessories

Agent Hardware

- Notebook based on Pentium II or above CPU
- 512 kB of RAM
- 20 GB HDD
- Interface for Phone connection (serial, USB, BlueTooth…)
- Cellular Phone
- Measuring set

4.2 Requirements on the Software

Server software is platform independent and can be run on both Windows and Linux system platform. Software for agent collecting measured times must be run on the Windows platform at this stage, because it has not been designed platform-independently yet. Other agents are implemented as web-based applications so they can be run on both Windows and Linux platform.

Server Software

- Any operating system supporting the software listed below
- Apache Web Server
- PHP interpreter
- MySQL database engine
- Control center application
- Web browser (MSIE, FireFox, …)

Agent Software

- Windows (for measuring agent) or another operating system (for other agents)
- Measuring application (measuring agents only)
- Web browser (MSIE, FireFox, …)

4.3 Message Transmission

Data are transmitted in the form of messages. Each message consists of its unique ID, Agent ID and information type. Message used to transport racer's data includes ID of racer also. Most types carry additional payload with the relevant information (see Fig. 5). The message is passed as URL parameters through HTTP request:

```
http://server/app/?mid=123&aid=2&t=1&rid=14
```

The server response has plain text format, the text consist of two rows. First row contains the status code (OK for all right status or ERR for error) and the second one contains some additional information. Client part is responsible for correct data transmission. If the server response includes error code, the client will try to transmit the message next time. All unsent messages are stored on client side to be available until they will be successfully sent.

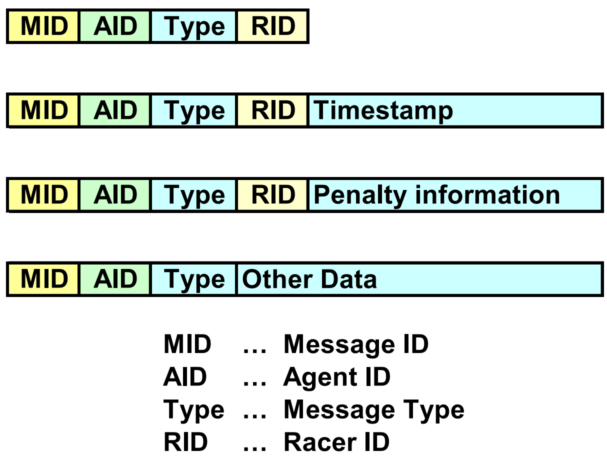

Fig. 5 – Message Types

The best way how to improve the transmission is to use SOAP architecture to transmit data as a small object. This standard provides some additional features usable for the secure communication including authentication methods. Other profit can come from precise service-interface which SOAP standard defines. It allows using other third party system components designed for SOAP communication.

Both the server and client part of SOAP enabled system can be easily realized by standard PHP classes reachable in standard Apache Web Server module package.

In addition, new types of cellular phones include capability of direct communication with web server using some Java application. It can provide possibility to design next generation of agents with ability to pass data directly.

5 Database Background

The relation database is the optimal tool to store all the system data. The MySQL database engine used in the system provides all the services needed for the best scalability and operability with the PHP interpreter compiled within Apache Web Server.

The system uses 9 tables for data storage. Relational model describes the structure of data records and the data relations (see Fig. 6).

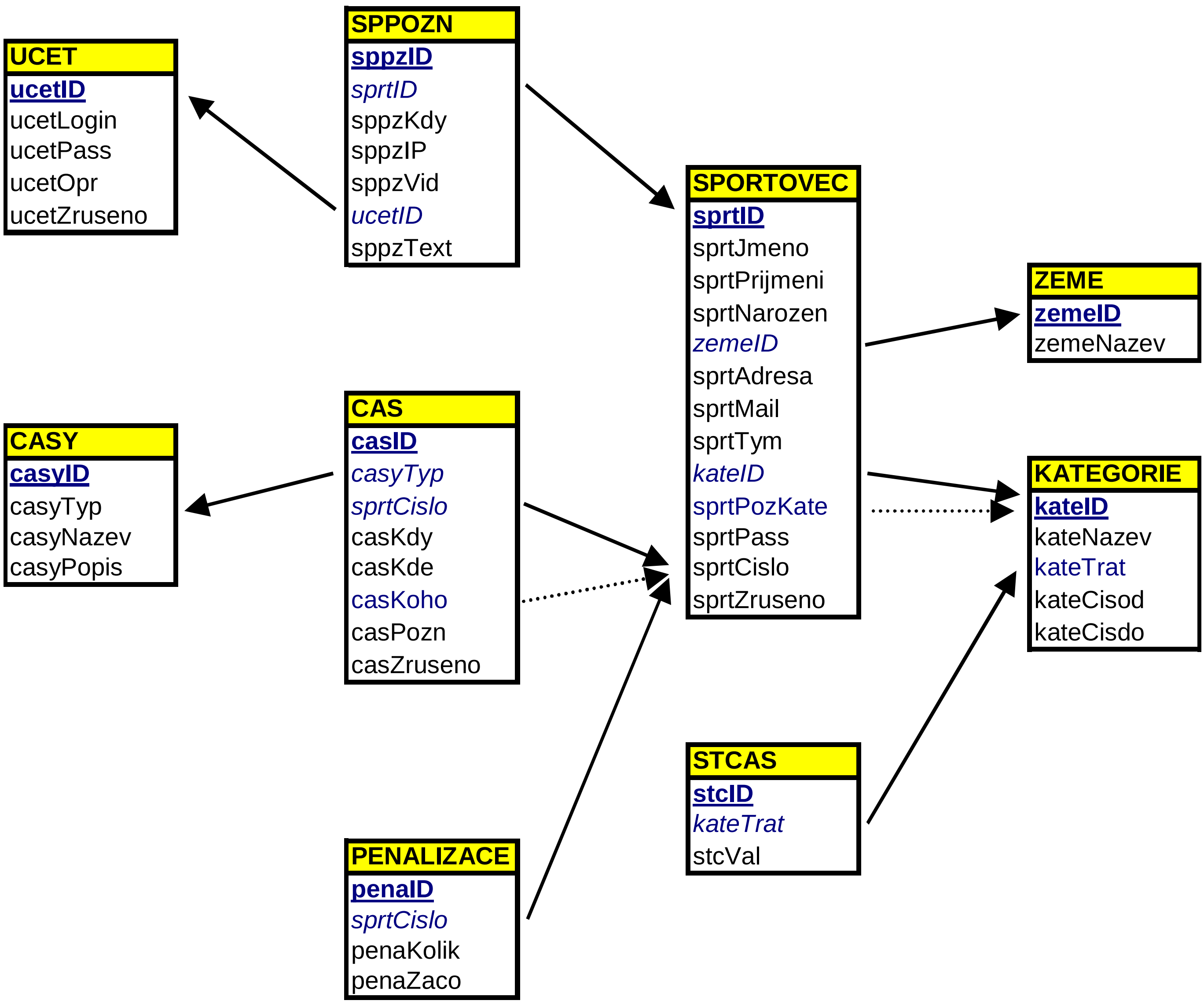

Fig. 6 – Relational Model

5.1 Data Model Description

The data are stored and organized in the tables. There are two main groups of tables. The first group contains data of admin and operator accounts, data of the racers, and race category definitions. In the second group of tables the measured data are stored including additional information, penalization and other race conditions.

5.2 Table Descriptions

The data model entities stored in the database are arranged in tables by the following scheme.

5.2.1 Table UCET – admin and operator accounts

Description:

The table stores accounts of person privileged to racer records management.

Fields:

ucetID – unique record ID

ucetLogin – unique login name

ucetPass – admin and operator password

ucetOpr – account privileges

ucetZruseno – disabled account flag

5.2.2 Table SPPOZN – racer data editing log

Description:

The information about privileged user(s) activity is logged in this table. Each action (login, logout, racer data editing, …) is stored including IP address and timestamp of the operation.

Fields:

sppzID – unique record ID

sprtID – foreign key, racer unique ID

sppzKdy – action timestamp

sppzIP – admin/operator IP address

sppzVid – public visibility of the action

ucetID – foreign key, admin/operator unique ID

sppzText – description of the logged action

5.2.3 Table SPORTOVEC – racer data

Description:

This table stores racer personal and race data. Racer must fulfill a registration form situated on the system web. The required data are full name, date of the birth, country and full address, e-mail address, selected race category and a password for record editing. Optional the name of a team can be entered. Admin or operator places the racer in the selected category and the racer's number can be assigned after the personal data verification.

Fields:

sprtID – unique record ID

sprtJmeno – first name

sprtPrijmeni – last name

sprtNarozen – date of birth

zemeID – foreign key, country ID

sprtAdresa – racer address

sprtMail – racer e-mail address

sprtTym – racer team

kateID – foreign key, selected category

sprtPozKate – foreign key, required category

sprtPass – racer data editing password

sprtCislo – assigned racer number

sprtZruseno – record validity flag

5.2.4 Table ZEME – country codes and names

Description:

The table stores country codes and the appropriate names of the countries.

Fields:

zemeID – unique record ID, country code

zemeNazev – country name

5.2.5 Table KATEGORIE – category descriptions

Description:

The category description is stored in this table. There are a name of the category and the track type code. The range of the racer numbers for this category is included.

Fields:

kateID – unique record ID

kateNazev – name of category

kateTrat – track type code

kateCisod – minimal racer number in category

kateCisdo – maximal racer number in category

5.2.6 Table STCAS – start timestamps

Description:

There are two tracks with different time of start. The timestamps are stored in this table.

Fields:

stcID – unique record ID

kateTrat – foreign key, track type code

stcVal – timestamp of the race start

5.2.7 Table CASY – timestamp types

Description:

This table stores timestamp type descriptions. There are some types of timestamp along the tracks, each one must be recognized by the type acronym. Additional data of the track conditions are tagged with specific timestamp type acronym.

Fields:

casyID – unique record ID

casyTyp – unique timestamp type acronym

casyNazev – timestamp type name

casyPopis – timestamp type description

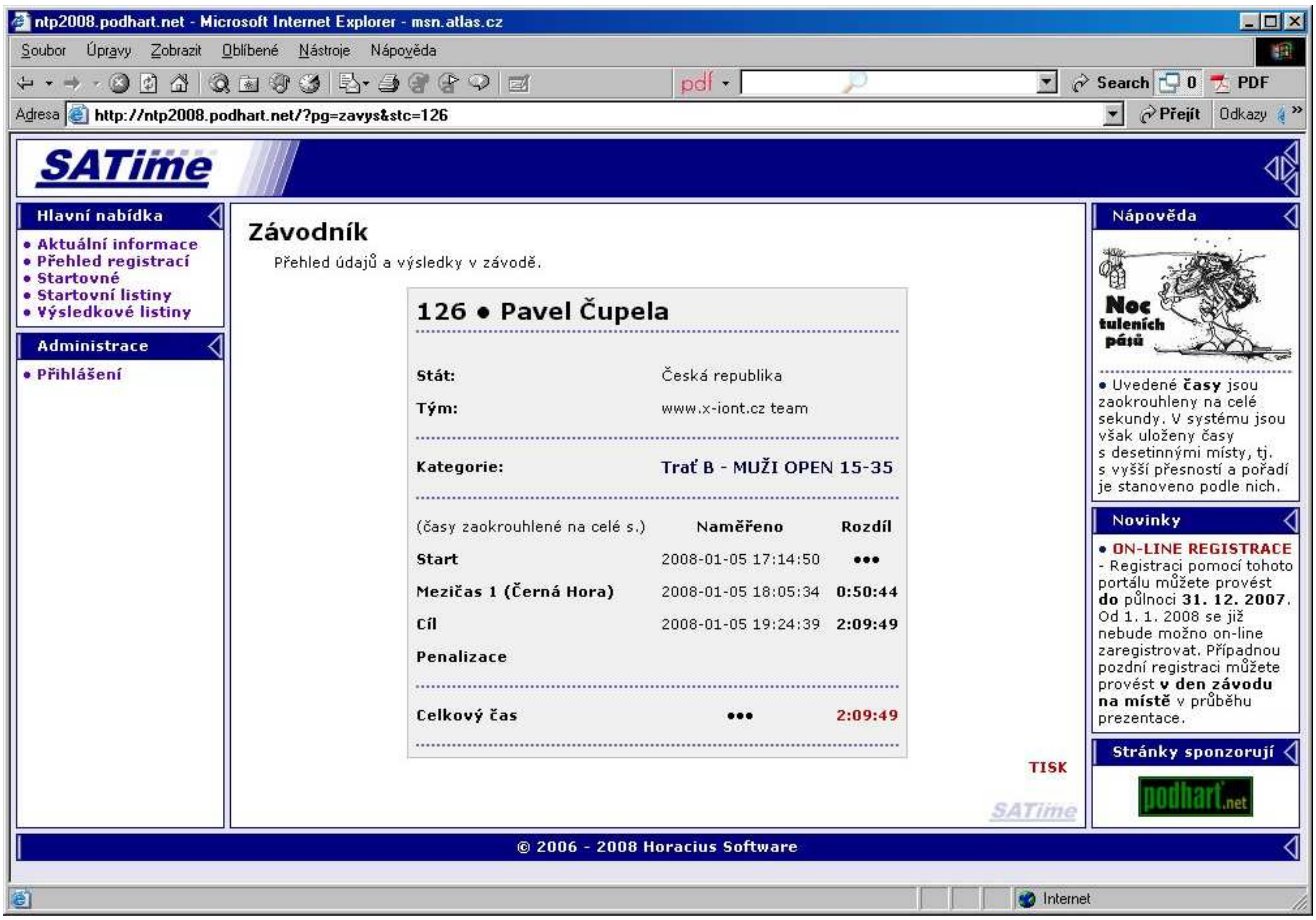

Fig. 7 – Result Presentation Interface

5.2.8 Table CAS – timestamps

Description:

This table stores timestamp type descriptions. There are some types of timestamp along the tracks, each one must be recognized by the type acronym. Additional data of the track conditions are tagged with specific timestamp type acronym.

Fields:

casID – unique record ID

casyTyp – foreign key, unique timestamp acronym

sprtCislo – foreign key, assigned racer number

casKdy – timestamp value

casKde – measuring agent identification

casKoho – recognized racer number

casPozn – additional information

casZruseno – record validity

5.2.9 Table PENALIZACE – penalization data

Description:

The additional time in seconds as the penalization is stored in this table. The record includes a value and description of the penalization.

Fields:

penaID – unique record ID

sprtCislo – foreign key, racer number

penaKolik – value of penalization in seconds

penaZaco – penalization reason description

5.3 Data Processing

The collected data processing is provided by the PHP application using MySQL database engine connection module in cooperation with web-based administration interface. The data management functions are designed by adaptive technology for relational data set as described in [6].

The measured data are received by application from agents and stored in database tables. The processing part of the application manages records, evaluates conditions and calculates results. Web-based presentation online availability is provided by the application (see Fig. 7).

6 Racer Identification

Next problem is connected to the task how to recognize identity of racers during the race. Each racer must be identified quickly, correctly and with a minimal effort. As it is written above, the racer's unique identifier (RID) is included as a mandatory field in some types of used messages so in such cases it must be available.

6.1 Present Way of Identification

In the current version of the system the racer identification is managed by referees at the checkpoints or by means of RFID devices on the most important locations (e.g. finish). But the RFID technology demands very expensive equipment and it cannot be used in places where the power supply is unavailable.

The identification of racers managed by referees is based on visual identification. Racers have number on their clothes. Referee must enter racer's starting number into the agent software running on a notebook. This way may be inaccurate because of a poor number readability in the darkness. Therefore it can be used only at special checkpoints where the count of passed racers is important but racer identity is marginal. These checkpoints serve to security purposes only – to identify at which parts of the track there are no racers.

6.2 Design of a New Identification Way

The current way of racers identification is not satisfactory. One of the possible solutions of the above stated problems is to use a technology based on a speaker's recognition by voiceprint to acquire identities of racers. The basic idea is that racers enter their voiceprints to the system before start of a race. Then, the system processes voiceprints, obtains the voice characteristics and stores them in database. When racers come to the checkpoints, their voice will be compared with stored voice characteristics and racers will be recognized.

The voice identification technology provides three types of speaker identification [7]:

- Text-dependent – the speaker is identified by known phrase
- Text-prompted – the speaker is asked to say one of previously defined set of words
- Text-independent – the speaker is identified by random word or sentence

The text-prompted form is a special modification of the text-dependent form using more known phrases but the prompting equipment realization is not appropriate for this case of use. The text-independent form of speaker recognition would be usable excellently but the process of recognition of random text is very difficult and time-consuming which makes this form of recognition unsuitable.

For functionality required during the race utilization of the text-dependent form of identification is an appropriate solution. Known phrase can be the for example racer's name.

6.3 Main Benefit of New Design

The racer recognition by means of voice characteristics does not require any special hardware, only a microphone is needed. The technology is software-based therefore no additional equipment is necessary. It will be easy to build more checkpoints on the track. If the number of checkpoints increases, the racers' positions will be determinable more accurately thus the security of racers will be enhanced. If the racer loses his/her way, size of the area to be looked through would be smaller.

6.4 Problems of New Design

The voice identification brings several questions and problems. The newly designed solution must work as quickly as the present system and amount of transmitted data must remain small. When a racer comes to a checkpoint, he/she will be out of breath and voice characteristics will be distorted. So the voice processing must be able to handle the distortion. Therefore the main problems to solve are:

- Where to process the voice – at the client or server side?
- Which data should be transmitted – voice sample or processed characteristic?
- How to handle the voice distortion?

If the voice processing is at the client side, only the characteristics will be transmitted, the amount of transmitted data will stay small but the client software will be more demanding. On the other side, if voice processing is done on the server side, the client software will remain easy but it will be necessary to transmit the whole voice sample. It means amount of transmitted data will increase significantly.

The handling of distortion may be very difficult. But some features of voice characteristics are not distortion-sensitive. Fundamental frequency or pitch period is robust to noise and channel distortions [8]. Of course, many tests must be realized before this solution can be used as a reliable way of racers' identification.

7 Conclusion

The web-based applications are the solutions of the future. Many systems are platform dependent requiring specific operating systems and specialized software bundles installations. The next way is to provide independent system and application and lease it instead of the buy-and-install software.

It is easy to develop some sophisticated system with a sufficient amount of money. The low budget conditions bring many complications and limitations. In this very specific case including hard-accessible mountain terrain are the limitations cardinal.

However, the proposed system solves most of the limitations, meets the given requirements and can be implemented with low costs. The most important advantage of the proposed solution is utilization of easily accessible technologies and communication tools, which are platform-independent, and utilization of free and open source software. Regardless of this choice the proposed solution meets next important requirement – utilization in hard-accessible mountain terrain where wireless connection is not available and providing data and results online or with a short delay only.

Proposed automation of racers' identification by means of speaker's recognition by voiceprint can significantly improve safety of racers. This technology allows to keep more checkpoints along the track which can identify racers so more accurate information about racers' position on the track would be available. This information can decrease are to search for lost racers. It is an important issue especially because of the race conditions (night and quite cold weather) so racers' lives can be in a big danger. Next advantage of proposed racers' identification is decreasing of number of referees. For the future, more detailed design and testing of the proposed way of racers' identification is planned.

References:

[1] Network Working Group *Hypertext Transfer Protocol - HTTP/1.1*. W3C/MIT, 1999. On-line at http://www.w3.org/Protocols/rfc2616/rfc2616.html

[2] Network Working Group *Simple Object Access Protocol (SOAP) 1.1*. W3C/MIT, 2007. On-line at http://www.w3.org/TR/soap/

[3] Ahmad, M., Iqbal J., Quart-Ul-Ain, Ghazal, S. Real Time Fleet Monitoring and Security System using GSM Network, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol. 5, Issue 11, 2006, pp. 2109-2114.

[4] Mohajerani, M., Moeini, A. Using Enterprise Architecture Framework to Design Network Security Architecture, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Vol. 3, Issue 2, 2004, pp. 688-693.

[5] Alter, S. *Information Systems: Foundation of E-Business*. 4th edn. Prentice-Hall, Upper Saddle River, 2002.
[6] Pliuskuviene, B., Adomenas, P. An Adaptive Technology for Processing Data Relational Sets and its Application, *WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS*, Vol. 4, Issue 4, 2007, pp. 648-654.
[7] Campbell, J. P. Jr. *Speaker Recognition: A Tutorial*. Proceedings of the IEEE, Vol. 85, No. 9, September 1997.
[8] Feng, L. *Speaker Recognition*. Kgs. Lyngby 2004.