UNIVERSITY OF TROMSØ UIT

FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

Ruoksat

A system for capturing, storing and presenting the digital footprints of coach knowledge and execution

Kim-Edgar Sørensen

INF-3981 Master's Thesis in Computer Science
June 2013

Abstract

Soccer teams and athletes are constantly looking for possibilities to gain advantages over opponents. In soccer your next opponent is analyzed down to the smallest details to find weaknesses and strengths. All this to be able to take advantage of your opponents weak points and handle their strengths. An example of a typical weakness is a team playing 4-4-2 and gives up space between the lines (back-four and midfield). An strength can be finding this space, typical with a number 10 player.

There are several ways to gather information about your opponent. From looking through whole matches to advanced tools, which can highlight key information for you.There are two main aspect of the analysis process; First you need to gather the information and secondly is how to present the information, usually for the coaching staff.

Acknowledgements

Table of Contents

List of Acronyms

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Soccer teams and athletes are constantly looking for possibilities to gain advantages over opponents. In soccer your next opponent is analyzed down to the smallest details to find weaknesses and strengths. All this to be able to take advantage of your opponents weak points and handle their strengths. An example of a typical weakness is a team playing 4-4-2 and gives up space between the lines (back-four and midfield). An strength can be finding this space, typical with a number 10 player.

There are several ways to gather information about your opponent. From looking through whole matches to advanced tools, which can highlight key information for you.There are two main aspect of the analysis process; First you need to gather the information and secondly is how to present the information, usually for the coaching staff.

1.2 Problem definition

This thesis will develop a system complementing the Muithu and Bagadus systems. Focus will be on soccer opponent analytics, where a data repository need to be developed capturing important events relevant for this type of analytics. Specially we want to identify the breakthrough players in a attack. Also a user interface component providing the core information about the opponent should also be developed.

1.3 Interpretation

Our project is that providing an infrastructure for capturing events like attacks that leads to an attempt on the opponent goal, and presenting this information through an user interface with graphs and illustrations. We are interested in how long it typically takes to manually capture all relevant events from a single match. For this we need to develop a data model that reflects what we want to capture from each attack. This for being able to do relevant queries on the data afterwards to gain insight.

First a requirement specification shall be developed. Then a prototype and that fulfills the requirements. At last the system shall be evaluated by specialize in the domain of soccer.

The design and development processes will be performed in collaboration with staff of the Norwegian soccer club Trms IL. An end-user comparison of the system against currently available tools will perform a final evaluation, and the result of this evaluation will be used to conclude the thesis.

1.4 Methodology

The final report of the ACM Task Force on the Core of Computer Science [1] divides the discipline of computing into three major paradigms:

- *Theory*: Theory: Rooted in mathematics, the approach is to define problems, propose theorems and seek to prove them in order to determine new relationships and progress in computing.
- *Abstraction*: Rooted in the experimental scientific method, the approach is to investigate a phenomenon by stating hypothesis, constructing models and simulations, and analyzing the results.
- *Design*: Rooted in engineering, the approach is to state requirements and specifications, design and implement systems that solve the problem, and test the systems to systematically find the best solution to the given problem.

For this thesis the design process seems to be the most suitable out of the three paradigms. The design process consist of 4 steps and is expected to be iterated when tests reveal that the latest version of the system does not satisfactorily meet the requirements.

- *State requirements and specification*: A need or problem is identified, researched, and defined.
- *Design and implement the system*: Data models and a system architecture are designed. Prototypes are implemented.
- *Test the system*: Assessment and testing of the aforementioned prototypes.

1.5 Outline

- *Item 1*: Presents some background information related to the thesis
- *Item 2*: Describes the requirement specification
- *Item 3*: Describes the general system model

Chapter 2

Related work

In this chapter we present some background related to creating a multi-media data repository. There are two main approaches: capturing data manually often by human annotating it, and automatic capturing. Automatic may or may not have a substep where a human has to filter/annotate the data.

2.1 Types of analyse

In soccer there are several phases where you use analytic to help you gain insight.

2.1.1 Prematch

Pre-match you use analytic to find weaknesses in the opponents team, on individually bases or the team as whole. You look at your team matched up against your opponent. A typical situation is that the manager gets an video summary of the opponent highlighting the opponents strengths and weaknesses. The video summary is often made up by the coaching staff who may use tools like Interplay.

Typically in TV you have pundits bringing you analytic of key battles during the build up to the game.

2.1.1.1 In match

During match the coaching staff continuously analysis the match and makes adjustment. Of course it is the players who makes all the decisions in the end, but the coach is the boss and most of the time players listen to what hes says. An adjustment to the formation can potentially be the tipping point in the game.

Troms IL uses a system NAME INSERT that lets you annotate sequences of a game with entity's like player, comment. This information will then be time synchronized with the video feed. Later, like in the break or in the game even, you can search on entity's to get the corresponding video feed. As the system is available on an tablets players can in the middle of the match come to sideline to see a involvement. It can be anything that is tagged like a player involvement to a team move.

Using data gathered from sports data company like Opta you can get statistics live during the game. Its popular in TV to show statistics like ball possession percent, how far players have run or passes played to mention a few.

2.1.1.2 Post match

During the post match the coach team goes through the game to evaluate the team performance. This is valuable as you get very concrete information about good and bad.

2.2 Manually capturing

2.2.1 Opta

One of the big players Opta uses manual input to create their data repository. They have editorial teams across the world that captures data manually for the most popular soccer leagues. For example to capture statistics for one match, 3 humans have to be involved to be able to annotate all data. The data is captured via an application specifically created for the purpose of capturing data as quick and easily as possible. The editorial teams of Opta need to be able to identify a player, registrate a pass his made or a tackle, in a very short time to be able to keep up with the pace of the game. They study things like which shoe color a player has to be able to quickly identify the player.

Opta capture all types of actions like passes, type of pass, attacks, and interceptions. For each action they log they add a series of description tags like pitch coordinate, player, team and time-stamp. For every single pass they registrate if it was a through ball, normal ball or even a headed flick on from a long ball. For shots they registrate the foot it was kicked with, if it was a volley and so on . All this is done while the match is playing. About 1600 individual events are recorded in a standard match.


```
<Event id=”290575408” event_id=”5” type_id=”1” period_id=”1”
min=”0” sec=”5” player_id=”20856” team_id=”810” outcome=”1”
x=”44.6”y=”61.1” timestamp=”2007-08-12T13:00:24.827”
last_modified=”2007-08-12T13:00:25”>
<Q id=”1774596260” qualifier_id=”141” value=”91.6”/>
<Q id=”1429253465” qualifier_id=”140” value=”49.9”/>
<Q id=”1084400575” qualifier_id=”56” value=”Back”/>
</Event>
```


Here is an example of an event registrated in the Opta database. The event has a series of qualifiers describing it. Except from the obvious as timestamp and last modified dates we see that the player id, team id, time of event, x and y coordinates and the outcome of the event are registrated. Also we see that some extra details are included. In this example it maps to a pass from [44.6, 61.1] to [91.6, 49.9].


```
<Q id=”1774596260” qualifier_id=”coordX” value=”91.6”/>
<Q id=”1429253465” qualifier_id=”coordY” value=”49.9”/>
<Q id=”1084400575” qualifier_id=”56” value=”Back”/>
```


2.2.1.1 PROZONE/AMISCO

PROZONE3 is another system that tries to track players. Their data capture system incorporates 8-12 cameras, which is strategically positioned throughout the stadium so it covers 100 percent of the ground with some redundancy in case of a faulty component. They also incoperate the TV-camera feed, which obivously always follows the ball. All cameras are hooked into one server and uploaded at the end of the game before sent to undertake the tracking process.

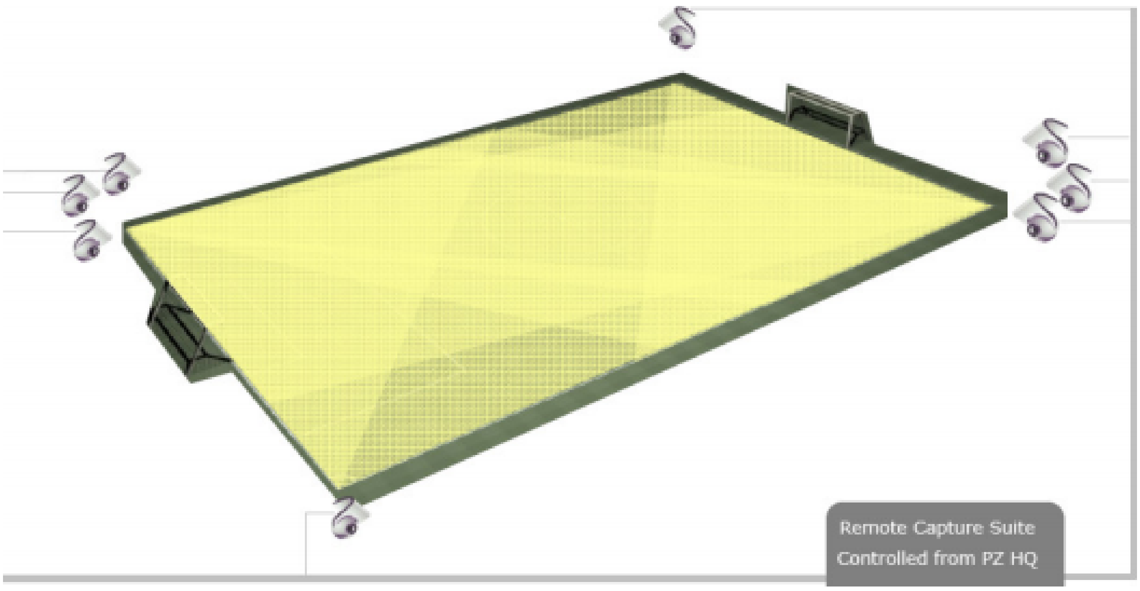

Figure 2.1: The PROZONE camera system illustrated

The coding and tracking process of the players movement and actions is a manual process at present time. The process of knowing where every players is and doing is still to be seen done by a software completely alone without human guiding. There is a large department dedicated for post processing the game. With the help of the many cameras with different camera angles the team track players and on the ball events. Events and players gets an positional tag by using the pitch dimensions and coordinates. As different football grounds houses different pitch sizes the pitch dimensions has to be taken into account by calibrating the cameras (ASK SIMON). How do you know how far a player has run? This is calculated from the events and notating where players has run combined with the known pitch size and coordinates. The whole input process is done in a own software which helps you minimize the amount of manual work. The software follows rules created by basic machine learning algorithms to validate and verify the input. A simple example would be if the ball goes out of play the system will now that the next event will be a throw in, corner kick or goal kick [**?**].

They claim to be able to track every movements of every player on the pitch every 10th of a second without using 3 any physical equipment on the players. When it comes to accuracy Di Salvo et al. [7] conducted an empirical evaluation of deployed ProZone systems at Old Trafford in Manchester and Reebook Stadium in Bolton, and concluded that the video camera deployment gives an accurate and valid motion analysis. The data is after a match available through the PROZONE3 interface.

2.2.2 Automatic capturing

SAP places sensors in shin guards, clothing, and in footballs. The data can be applied to individual and team movement profiles to track distances, speed averages, ball possession, player tendencies and more .

A similar system is the ZXY sport trackingsystem. The system is in used by premier league soccer teams in Norway, including Troms IL. Data captured is stored in Sybase databases with each match requiring about 500-700MB storage The players have to wear a belt around their waist for the system to work. The ZXY system is able to track the players movement very detailed with an accuracy of 0.5m. It has a resolution of 20 samples per second. It relies on a radio-based signaling substrate to provide real-time high-precision positional tracking, including acceleration and heart rate. The home arena for Troms IL, Alfheim, is currently equipped with 10 receivers . A receiver tracks an specific area of the soccer field and combined they cover the whole pitch with some redundancy areas. The communication from the belt to the receivers goes on a 2.45-5.2 G Hz frequency radio signal. To compute the positional data the stationary radio receiver uses an advance vector based processing of the received radio signal. The data is aggregated and stored into a relational database.

Including the positions of the players the ZXY also gives you the step frequency and speed.

2.2.3 Wrap up

The main problem with tracking systems that uses physical sensors is that usually only one of the teams wears the sensors. This limits the functionality of the system as a opponent analysis system. You only get data for one team.

On the other hand you have the manually systems that requires some human annotation. These system are able to track both teams. As they rely on human annotation of some degree they get more rich data as well. This includes pass type, key-passes, tackles, interceptions and so on.

Figure 2.2: The PROZONE software - individual player analysis gives you statistics of performance over time

2.2.3.1 Sensor based

2.2.4 Presenting data

Most presenting of data is based around single matches. Figure 2.2 shows an example of how FourFourTwo presents data from a match. They use statistics from Opta.

2.2.4.1 Prozone

PROZONE comes with several softwares to illustrate the data. The most relevant is the opposition analysis system.

2D animation Single player analysis Team analysis Pressing analysis Success/direction Player tempo Passing movements Receving the ball Player events

On individual level you can get basic tactics like shots, total passes, passing success, crosses, shots on target, balls received, tackles, fouls.

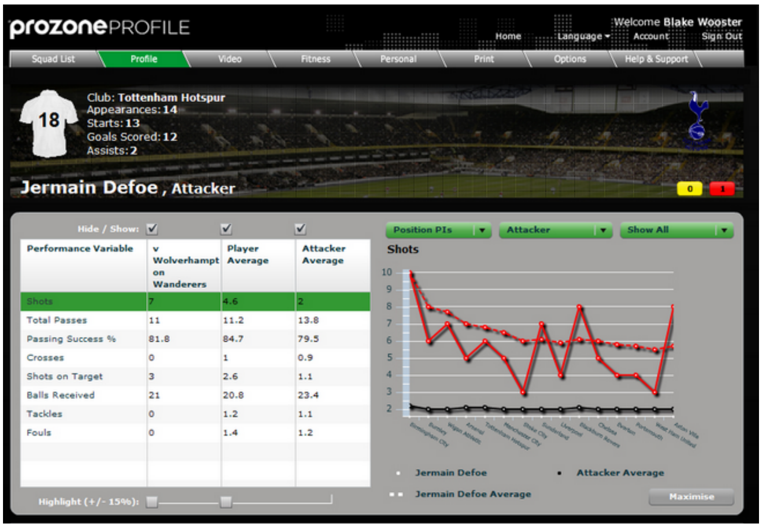

Figure 2.3: The PROZONE software - individual player analysis gives you statistics of performance over time

Doing queries on the data can give you all sprints for a certain player. Players in certain areas of the field have more intensive sprints when they first are involved thus are more vulnerable to injuries. Knowing the actual physical load on players can prevent injuries by regulating the training intensity and amount of time on each exercise.

2.2.4.2 ZXY

ZXY provides you with a 3D graphic interface. This interface lets you see the players action in real time by reading the data stream to reproduce the players action. The data is streamed in real time into the database as the match goes on. While watching you can produce timestamps of different events and produce manual input which is time synchronized with the automatic data. Naturally you can build your own software on top of the Sybase database. Troms IL in collaboration with University of Troms has made several systems to complement the ZXY software. Muithu and Bagadus.

[?]

Figure 2.4: The ZXY software - tracking of players gives you information of distance covered, average speed and max speed. You also get a heat map of the players movement on the field.

Chapter 3

Requirement Specification

This chapter outlines the requirement specification of the system. This section describes the requirement analysis process. The process of gathering the requirements was done in collaboration with Troms IL.

3.1 Overview

The requirements evolved during the process. First a requirement specification was designed from our perspective. We looked at the different analyze systems out there and as mentioned in chapter 3 a good system for locating key players in opponent teams was lacking. Rather than going very wide providing all kind of analyses we narrowed it down to a very concrete system. A system that tries to to for many things may fell between two chairs and not providing anything real insight. The time spent on each feature is reduced and so the quality.

The imagined system shall give you the key players in the offensive play of a given opponent soccer team. You shall be able to search on teams and individual players. Additionally you shall be able to see which areas of the pitch players are creating goal chances from. The system also needs a way of capturing data. This shall be an interface that enables you to store successful attacks for any match.

What shall be captured?

A domain model for the captured data is crucial to set early and don't change it radically. Initially we wanted to build a database of all the matches in the

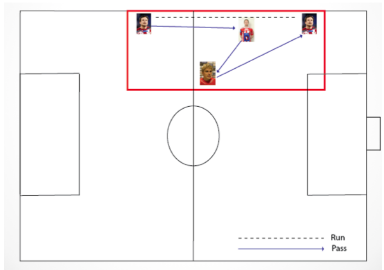

Figure 3.1: Example of one of the illustrations displaying key players and how they combine after a search on a specific team

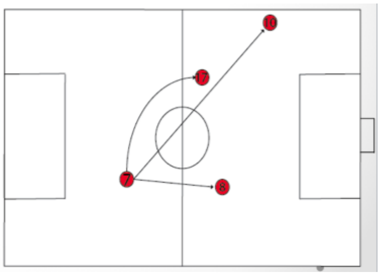

Figure 3.2: Example of one of the illustrations displaying typical passes for a player

Norwegian premier league. From each match every successful attack a team makes should be captured. From the attack started you registrate where the attack started, every pass with from position to the new position, and type of attack. At last if there is a breaking point in the attack this should be captured. The breaking point of the attack is stored with a breakthrough player and what type of breakthrough it was.

Definition of breakthrough player: A player that does something extra that unbalance the other team. This can be a dribble past 1-2 players or a genius pass that opens deference of the opponents team.

First problem was how to divide the pitch into zones. As we are looking for which zones the breaking point of the attack this is crucial for the searches on the data captured later on. During the development of the system several types of dividing was presented to the coaching staff.

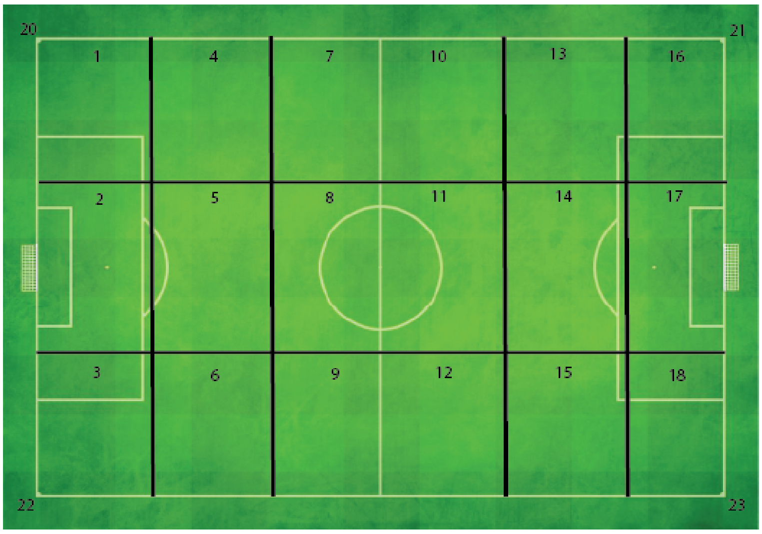

Figure 3.3: Dividing of pitch - the first suggestion had 18 zones. The team attacking attacks from left to right

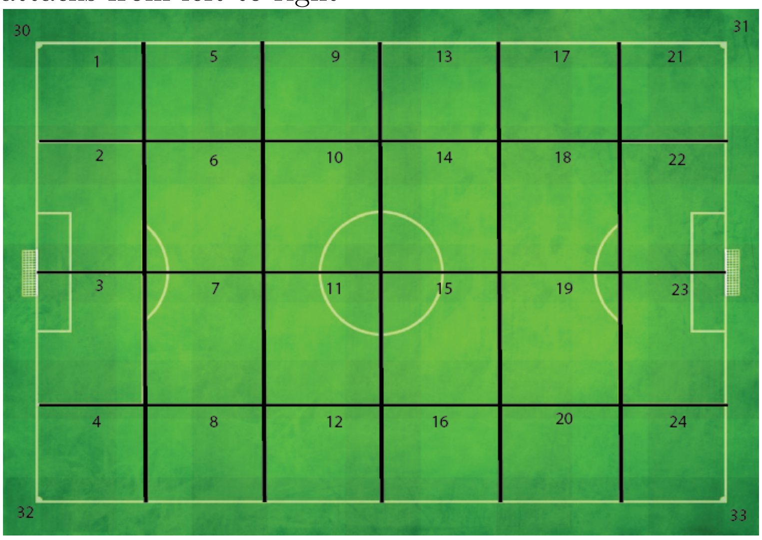

Figure 3.4: Dividing of pitch - the second suggestion had 18 zones. The team attacking attacks from left to right

Chapter 4

System Model

Chapter 5

Soccer Analytic toolkit

5.1 System Architecture

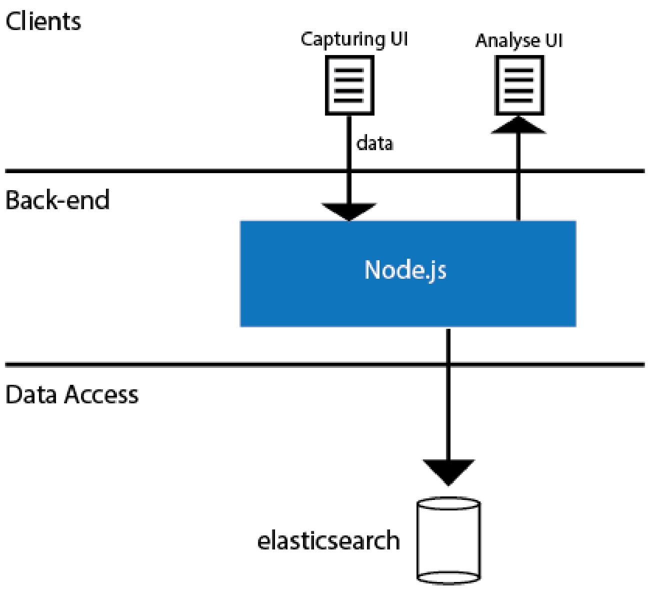

Figure 5.1:

The system architecture can be layered in to three layers; client, back-end, database. The control flow flows from the clients requesting something or inserting data, to the back-end and to the third layer, the database, and back up again in reverse order.

5.1.0.3 Interfaces

There is several interfaces but the most important are for capturing new matches and attacks, and the for analyze information. The back-end then again connects to an database to fetch or insert data. The request then returns up again in the stack on response from the database.

5.1.0.4 Back-end

The back-end is the middle layer between the client and the storage. Its main task is to serve the UI's to the clients, handle data insertions or queries by mapping them to database operations.

5.1.0.5 Storage

The storage layers task is to persist data and handle search queries on the data. It consist of several indexes.

5.1.1 Domain model

The domain model is based around matches.

```
{
    "hometeam" : "Troms ",
    "awayteam" : "Rosenborg",
    "score" : "1-0",
    "date" : '2013-15-09',
    "attacks": [
        {
            "time": 4,
            "touch" : 1,
            "team" : "Troms ",
            "breakthrough" : "None",
            "breakthroughPlayer" : "None",
            "typeOfAttack" : "D dball",
            "attackStart" : {
                "pos" : 17,
                "typeAction" : "Frispark",
                "player" : 403,
```

```
            },
            "passes": [
                {
                    "fromPlayer": 403,
                    "toPlayer": 393,
                    "fromPos": 17,
                    "toPos": 23,
                    "action": "CROSS"
                }
            ],
            "finish" : {
                "player": 393,
                "pos": 23,
                "action": "SHOTMISS"
            }
        },
}
```

5.2 Interfaces

The first page you are prompted with is the listing of all matches registered in the database. A click on match gives you details about that match and prompts you a interface for capturing new attacks if requested. Every field has to be submitted with an correct input value.

5.3 Implementation details

Table 5.1: Software stack

Web server	Node.js
Database	Elasticsearch
Client	web browser

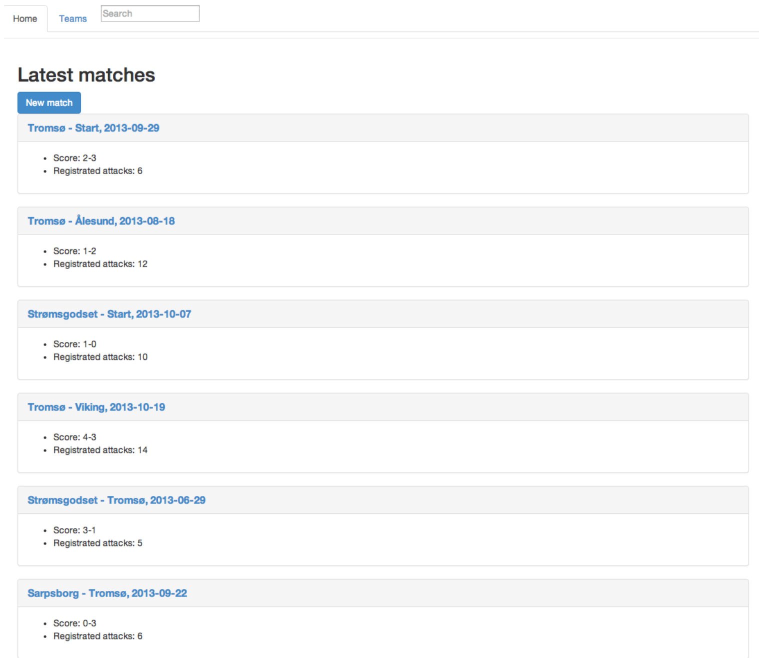

Figure 5.2: All matches registrated in the database are listed on this page

5.3.1 Storage

The data storage is an elasticsearch database. Elasticsearch is document oriented and works extremely well with JSON. As our server is built on JavaScript working with JSON is easy. JSON-objects can be inserted right into the storage and elasticsearch will map fields and value accordingly. Our data input is generated in the web browser which also uses JavaScript and could have been inserted right away into the database without any pre mapping. Elasticsearch takes advantages of embedded documents meaning we can store related data together. As an attack is usually made up of several passes you can store the passes as an embedded document inside the attack document so they can be retrieved in one query.

The main reason for using Elasticsearch is its search capability. In a single query you can get counted how many passes all player for a team has played and received, number of times all players has been the breakthrough-player, type of attacks, most used zones for passing and finishing and so on. This makes it very easy and efficient to do queries for analyses on teams and players. After a query you can return all data directly to the client for him to expose to the end user.

5.3.2 Back-end

The back end is the middle-ware between the clients and the data layer. It exposes a RESTful interface over HTTP for the client to communicate. A request coming in is transformed to a database query based on the resource it tries to access. On answer from the database the result is transformed before returning it to the client.

Similar if the client sends new data for a match the middle-ware inserts the data into the appropriate indexes.

5.3.3 Front-end

Front end is consist of a single page JavaScript application using Backbone.js as under-supporting framework. As it uses a MVC structure the models is responsible for AJAX communication with the back-end.

For a analytic toolkit to be useful a good UI is critical. Here several helper library is used to present the data. Highcharts.js is JavaScript library for illustrating graphs. A query on team generates a lot of statistics and rather than listing them up they are presented using charts. This also gives us the advantage of displaying several numbers for each player and plot it in the same graph. In the image below we show the number of times a player has been involved in all attacks, number of passes into the final third of the pitch and the number of times a player has been the breakthrough-player.

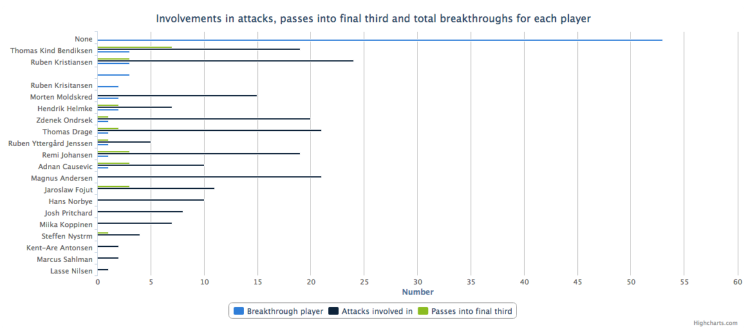

Figure 5.3: Shows how the passing statistic is illustrated on the client by using Highcharts.js

Positional data is created using the a new feature of HTML5, canvases. From the model you get all zones with a number that symbols shots taken from that zone. This is then plotted into the respective zones.

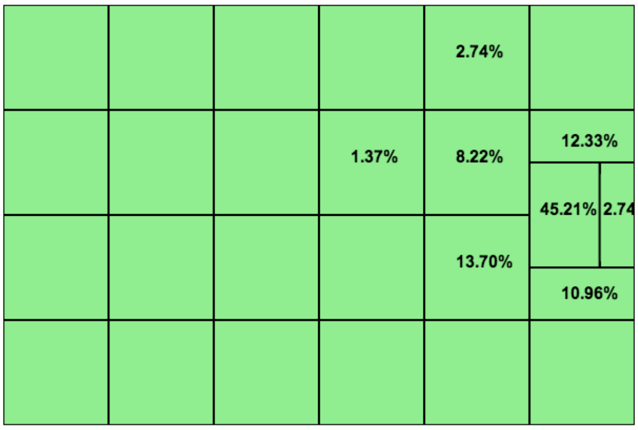

Figure 5.4: Illustrations of which zones the team has finished off their attacks from, with percent (team: Troms IL)

Backbone comes with a library Underscore.js that makes creating HTML pages with dynamic content easily. When you rending a new page on the site you can insert content retried from a model into the HTML and then render it.

5.3.3.1 Security

Secturity is not taken into convern. This means anyone getting into the page can post new match data.

Chapter 6

Evaluation and Results

6.1 Limitations

As the input is manual the current biggest limitations is humans.

The input is to some degree subjective for some data like identifying breakthroughs.

Chapter 7

Conclusion

This chapter presents our achievements, gives some concluding remarks and outlines possible future work.

7.1 Achievements

7.1.1 Concluding Remarks

7.2 Future Work

References

[1] D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. Computing as a discipline. *Commun. ACM*, 32(1):9–23, January 1989.

[2] Pl Haloversen Hvard D. Johansen, Svein Arne Pettersen and Dag Johansen. Combining video and player telemetry for evidence-based decisions in soccer.