

## Soccer Analytic Tool

---

**Simen Lomås Johannessen**

*INF-3983 Capstone Project in Computer Science  
December 2013*





# **Abstract**

Sports science is an increasingly hot topic and more and more soccer teams are investing time and money into strategic development of big data and sports science. All this to increase the players performance on the field and reduce the risk of injuries. Soccer teams and athletes in general 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 limits their strengths.

This project creates a system for capturing events relevant for opponent analysis and an interface providing the information for analyzing soccer opponents.



# Acknowledgements



# Table of Contents

<b>1</b>	<b>List of Acronyms</b>	<b>ix</b>
<b>2</b>	<b>Introduction</b>	<b>1</b>
2.1	Background . . . . .	1
2.2	Problem definition . . . . .	2
2.3	Interpretation . . . . .	2
2.4	Methodology . . . . .	3
2.5	Outline . . . . .	3
<b>3</b>	<b>Related work</b>	<b>5</b>
3.1	Types of analysis . . . . .	5
3.1.1	Pre-match . . . . .	5
3.1.1.1	In match . . . . .	6
3.1.2	Post match . . . . .	7
3.2	Capturing of data . . . . .	7
3.2.1	Opta . . . . .	7
3.2.2	Prozone . . . . .	8
3.2.3	Automatic capturing . . . . .	10
3.2.4	Wrap up . . . . .	10
3.3	Presenting data . . . . .	11
3.3.1	Prozone . . . . .	11
3.3.2	ZXY . . . . .	12
3.3.3	FourFourTwo . . . . .	13
<b>4</b>	<b>Requirement Specification</b>	<b>15</b>
4.1	Overview . . . . .	15
4.2	Capture . . . . .	16
4.3	Present . . . . .	17
4.4	Summary . . . . .	18
<b>5</b>	<b>Soccer Analytic toolkit</b>	<b>21</b>

5.1	System Architecture . . . . .	21
5.1.0.1	Interface . . . . .	21
5.1.0.2	Back-end . . . . .	21
5.1.0.3	Storage . . . . .	22
5.1.1	Domain model . . . . .	22
5.2	Implementation details . . . . .	24
5.2.1	Storage . . . . .	24
5.2.1.1	Indexes . . . . .	25
5.2.2	Back-end . . . . .	26
5.2.2.1	API . . . . .	26
5.2.3	Front-end . . . . .	26
5.2.3.1	Models . . . . .	27
5.2.3.2	Presenters . . . . .	28
5.2.3.3	Views . . . . .	28
5.2.3.4	Router . . . . .	30
5.2.4	Getting players and teams into the database . . . . .	30
5.2.5	Security . . . . .	30
<b>6</b>	<b>Demo</b>	<b>31</b>
6.1	Interfaces . . . . .	31
6.1.1	Mockup . . . . .	31
6.1.2	Implemented interfaces . . . . .	31
6.1.2.1	Home page . . . . .	31
6.1.2.2	Registering attacks . . . . .	32
6.1.2.3	Match view . . . . .	32
6.1.2.4	Team view . . . . .	32
6.2	Capturing process . . . . .	33
<b>7</b>	<b>Evaluation and Results</b>	<b>45</b>
7.1	Methods . . . . .	45
7.1.1	User Survey . . . . .	45
7.2	Experiments and Results . . . . .	46
7.2.1	Test data and Assessors . . . . .	46
7.2.2	SAT as a tool for opponent analytic . . . . .	46
7.2.3	SAT as a tool for identifying key players . . . . .	47
7.2.4	UI-performance . . . . .	47
7.3	Discussion . . . . .	47
7.3.1	Input . . . . .	47
7.3.2	Interfaces . . . . .	48
7.3.3	Scalability . . . . .	48

<b>8 Conclusion</b>	<b>49</b>
8.0.4 Concluding Remarks . . . . .	49
8.1 Future Work . . . . .	49



# Chapter 1

## List of Acronyms

**UI** User Interface

**AJAX** Asynchronous JavaScript and XML

**JSON** JavaScript Object Notation

**HTML** HyperText Markup Language

**API** Application Programming Interface

**REST** Representational state transfer

**HTTP** Hypertext Transfer Protocol

**MVP** Model View Presenter

**DOM** Document Object Model

**XML** eXtensible Markup Language



# List of Figures

3.1	A typical tweet from one of Opta's Twitter account . . . . .	6
3.2	The Prozone camera system illustrated. Taken from Prozone-sports.com . . . . .	9
3.3	Overview of the ZXY system with receivers placed at the stadium and players wearing sensors. Image from zxy.no . . . . .	11
3.4	The Prozone software - individual player analysis gives you statistics of performance over time. Image from prozonesports.com . . . . .	12
3.5	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. Image from zxy.no . . . . .	13
3.6	Illustration of all passes in to attacking third in the match Manchester United vs Newcastle United. Taken from four-fourtwo.com . . . . .	14
4.1	A screen capture showing some data Troms IL captured in a previous analytic project . . . . .	16
4.2	Illustrations of different breakthroughs we want to register .	17
4.3	Dividing of pitch - the first suggestion had 18 zones . . . . .	18
4.4	Dividing of pitch - the second suggestion had 24 zones . . . . .	18
4.5	Conceptual architecture . . . . .	19
5.1	Overall architecture of the system . . . . .	22
5.2	Final pitch dividing. More zones have been added to the penalty box area than the initially design. . . . .	24
5.3	Software stack . . . . .	24
5.4	Overview of the web servers API . . . . .	27
5.5	Architecture of the client side . . . . .	27
5.6	Shows how the passing statistic is illustrated on the client by using Highcharts.js . . . . .	29

5.7	Illustrations of which zones the team has finished off their attacks from, with percent. Team - Troms IL) . . . . .	29
6.1	A mockup of the main analytic page . . . . .	34
6.2	All matches registered in the database are listed on this page . . . . .	35
6.3	Interface listing up all attacks for a match . . . . .	36
6.4	Interface for registering an attack . . . . .	37
6.5	Interface listing all teams . . . . .	38
6.6	Main interface for information about opponents - key aspects . . . . .	39
6.7	Main interface for information about opponents - offensive play . . . . .	40
6.8	Main interface for information about opponents - defensive play . . . . .	41
6.9	Main interface for information about opponents all players . . . . .	42
6.10	Player view highlighting individual statistics . . . . .	43
6.11	Player view highlighting individual statistics . . . . .	44
7.1	Matches that have been captured and persisted into the database	46

# List of Tables



# Chapter 2

## Introduction

### 2.1 Background

Sports science is an increasingly hot topic and more and more soccer teams are investing time and money into strategic development of big data and sports science [7]. All this to increase the players performance on the field and reduce the risk of injuries. One of the pioneers in this field for a long time AC Milan established in as early as 2002 the MilanLab [5]. The goal for the program was to get better and more concrete information about the players physicality by collecting data over time of players performance. As AC Milan was one of the first to use big data, the use of big data today has exploded. Services like Opta and Prozone serve player statistics, heat maps, and video analytics of players performance. Fans get exposed to these statistics by clubs and broadcasting companies that use it actively in their coverage of soccer games. The awareness of statistics for clubs and fans has possibly never been higher than now [4].

Soccer teams and athletes in general 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 limit their strengths. The players should know what to expect from the opponent team. Detecting typical team plays and player movement of your opponent can help you prepare for the match.

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. Some of them are expensive as Prozone [8] which is a complete system for capturing and presenting information. For small soccer clubs with relatively small budgets this can be a expensive investment. Another system is Interplay Sports, which Troms IL uses for game analytic. A video feed from any match can be used as the input letting you manually tag and describe situations in the match. In a sport where the next soccer match usually is in 3 days tools for filtering out the useless data is valuable.

In the analytic process there are two processes we will look at; first you need to gather the information and secondly is how to present the information gathered.

## 2.2 Problem definition

This project will develop a system complementing the Muithu and Bagadus systems. Focus will be on soccer opponent analytics, where a data repository needs to be developed capturing important events relevant for this type of analytics. Especially we want to identify the key players offensive in a team. A user interface component providing the core information about the opponent should also be developed.

## 2.3 Interpretation

The 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 a user interface. We are interested in how long it typically takes to capture all relevant events from a single match, as this has to be done manually. A data model that reflects what we want to capture from each attack needs to be developed for being able to get relevant information for being used as a opponent analytic system.

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

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 project.

## 2.4 Methodology

The final report of the ACM Task Force on the Core of Computer Science [2] divides the discipline of computing into three major paradigms:

- *Theory*: Rooted in mathematics, the approach is to define problems, propose theorems and look to prove them in order to find new relationships and progress in computing.
- *Abstraction*: Rooted in the experimental scientific method, the approach is to analyze a phenomenon by creating 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 project the design process seems to be the most suitable out of the three paradigms. The design process consists of 4 steps, which are repeated if tests reveal that the latest version of the system does not 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.

## 2.5 Outline

- *Chapter 2*: Presents some related work to the project
- *Chapter 3*: Describes the requirement specification
- *Chapter 4*: Describes the design and implementation of the system
- *Chapter 5*: Gives a demonstration of the system
- *Chapter 6*: Evaluates and discuss the system
- *Chapter 7*: Concludes the project



# Chapter 3

## Related work

In this chapter we present some background related to analytic in the domain of soccer. We look at different types of analysis out there. This spans from pre-match to post-match. Then we go into how data is gathered and presented. For gathering of data there are two main approaches: manually often by human annotating it, and automatic capturing often by using sensors.

### 3.1 Types of analysis

In soccer there are several phases where you use analytic to help you gain insight. In this section we will look at the typical use cases of analytic. Not only soccer teams uses analytics, but also TV and fans widely uses statistics to build up under their arguments.

#### 3.1.1 Pre-match

Pre-match you use analytic to find weaknesses and strengths in the opponent team, on an individually level or as a team. You look at your team matched up against your opponent. A typical situation is that the manager gets a video summary of the opponent highlighting the opponents strengths and weaknesses. The coaching staff may use tools like Interplay to gather information and create the video summary. Other software like Prozone?? will also likely be used to break down the opponents offensive and defensive play.

Typically in TV you have pundits bringing you analytic of key battles during the build up to the game. A typical thing is to look at battles like wingers versus backs, midfield clashes or striker versus central defenders. The battles are often illustrated with statistics or with video clips highlighting aspects of the both players game that will be decisive.

Fans are analyzing the game all the time. Except from TV they get exposed to analytic via social media. Opta is company that has taken full advantage of social medias. Figure ?? is a screenshot of a tweet from one of the many accounts Opta control.



Figure 3.1: A typical tweet from one of Opta's Twitter account

### 3.1.1.1 In match

During match the coaching staff continuously analysis the match and makes adjustment. The players make all the decisions in the end, but the coach is the boss and sets the style of play. An adjustment to the formation can potentially be the tipping point that wins you the game.

Troms IL uses a system Muithu 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 at half time or even in the game, you can search on entity's to get the corresponding video feed. As the system is available on tablets players can in the middle of the match come to sideline to see a involvement that the coaching staff would like to show. It can be anything that is tagged like a player involvement to a team move that was executed perfectly.

Using systems like ZXY or MiCoach you can get real time information about a players performance during the match. Rather than assuming that a player is tiring you can get information metrics that tells you this. You get evidence that the player in fact is fatigued. In soccer you only have 3 substitutions. Making the right ones is crucial in an even match.

Using data gathered from sports data company's like Opta you can get statistics live during the game. It's popular in TV to show statistics like ball possession percent, how far players have run or the number of passes played and so on. As mentioned earlier Opta posts many of their statistics to their social media accounts for fans and media.

### **3.1.2 Post match**

During the post match the coach team goes through the game to evaluate the team performance. This is valuable as you get concrete clips about good and bad involvements on team or player level. You can highlight situations in the match to help players better understand tactical aspects. Interplay Sports is a system used today at Alfheim for this purpose. It is used for analysis of matches in a post-match scenario where you can tag situations in the video.

## **3.2 Capturing of data**

### **3.2.1 Opta**

One of the big players in the field of sport analytic 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 and other sports. For example to capture statistics for a single match they have 3 humans annotating. 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 in a very short time to be able to keep up with the pace of the game. They even study things like which shoe color a player.

Opta capture all types of actions like passes, type of pass, attacks, and interceptions to mention a few. For each action they log, they add a series of description tags like pitch coordinate, timestamp, which player and team. For every single pass they add descriptors if it was a through ball, normal ball or a headed flick on from a long ball. For shots they register 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.

This gives them a very rich database of events and a range of possible queries to run.

```
<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>
```

Above is an example of an event registered 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 registered. 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].

### 3.2.2 Prozone

Prozone is a video-based system that tries to track players in team sports [8]. Their data capture system incorporates 8-12 cameras, which is strategically positioned throughout the stadium to cover 100 percent of the ground, but with some redundancy in case of a faulty component. They also incorporate the TV-camera feed, which 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. When they system is installed it required some setup before it is operational; cameras need to be calibrated out from the pitch size. Figure 3.2 shows the camera location on a soccer stadium.

The coding and tracking process of the player's movement and actions is a sort of manual process at present time. First each camera feed goes through each own tracking process determining image co-ordinates and continuous trajectories for each player. The output from all the cameras is then combined into one dataset. Going into the algorithm for this is out of the scope of this project. In the final stage the manual work comes into the loop. There has to be a quality control group of operators dedicated for post processing the game. First the operators has to map the players identity and with the their corresponding start location. The operators will then follow the video feed checking that the identification of all players remains constant

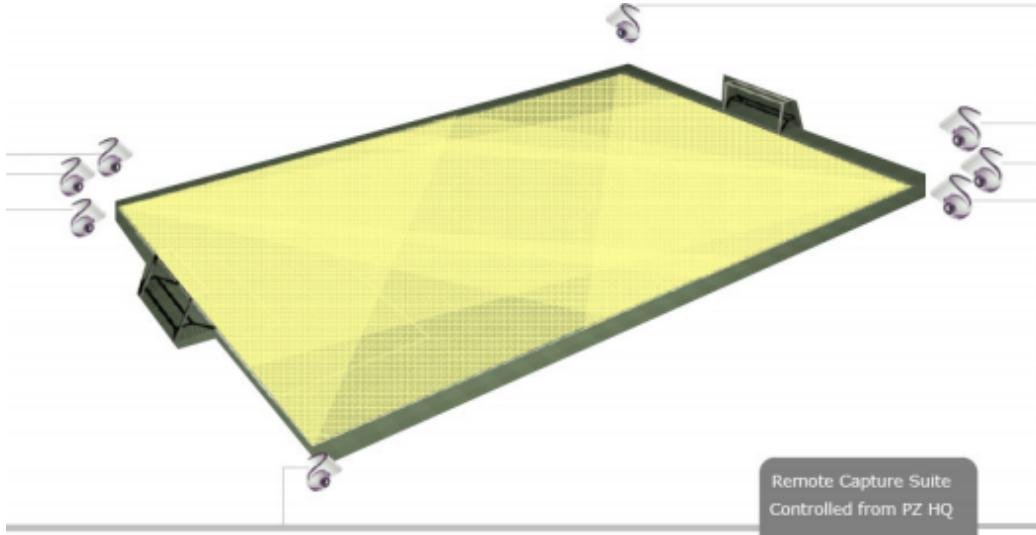


Figure 3.2: The Prozone camera system illustrated. Taken from Prozonesports.com

during the match. The tracking process may run into problems when two players collide or have any other physical contact as it becomes unclear who is who. To map video image co-ordinates to pitch co-ordinates Prozone uses computer vision homography calibration process.

The whole input process is done in a own software which helps you minimize the amount of manual work for the operators. 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.

Prozone claims to be able to track every movement of every player on the pitch with a resolution of 10th of a second. This without using any physical equipment on the players. Di Salvo et al. [1] conducted an empirical evaluation of deployed Prozone systems at Old Trafford in Manchester and Reebok 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 Prozone interface for analysis.

### **3.2.3 Automatic capturing**

A system that uses sensors is the ZXY sport tracking system <sup>1</sup>. The system is in used by premier league soccer teams in Norway, including Troms IL and Rosenborg BK. 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 be able to track their movements. 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. The technology behind it relies on a radio-based signaling substrate to provide real-time high-precision positional tracking, also including acceleration and heart rate. An installation of receivers is required for the system to work. The home arena for Troms IL, Alfheim, is currently equipped with 10 receivers. A receiver tracks a 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, speed and direction [3]

### **3.2.4 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 an opponent analysis system. You only get data for one team and if you have installed it at your stadium it captures only half of the matches.

On the other hand you have the manually systems that requires human annotation of matches. These systems have the advantage of being able to track both teams. As they rely on human annotation of some degree they get more rich data as well, but requires strict rules of the semantics of the metadata.

---

<sup>1</sup><http://www.zxy.no>



Figure 3.3: Overview of the ZXY system with receivers placed at the stadium and players wearing sensors. Image from [zxy.no](http://zxy.no)

### 3.3 Presenting data

This section gives a short overview of how some systems present opponent analysis.

#### 3.3.1 Prozone

Prozone comes with several softwares to illustrate the data. The most relevant for this project is the opposition analysis system. It includes features like 2D animation, single player analysis, team analysis, pressing analysis, success/direction, player tempo, passing movements, receiving the ball, player events.

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

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

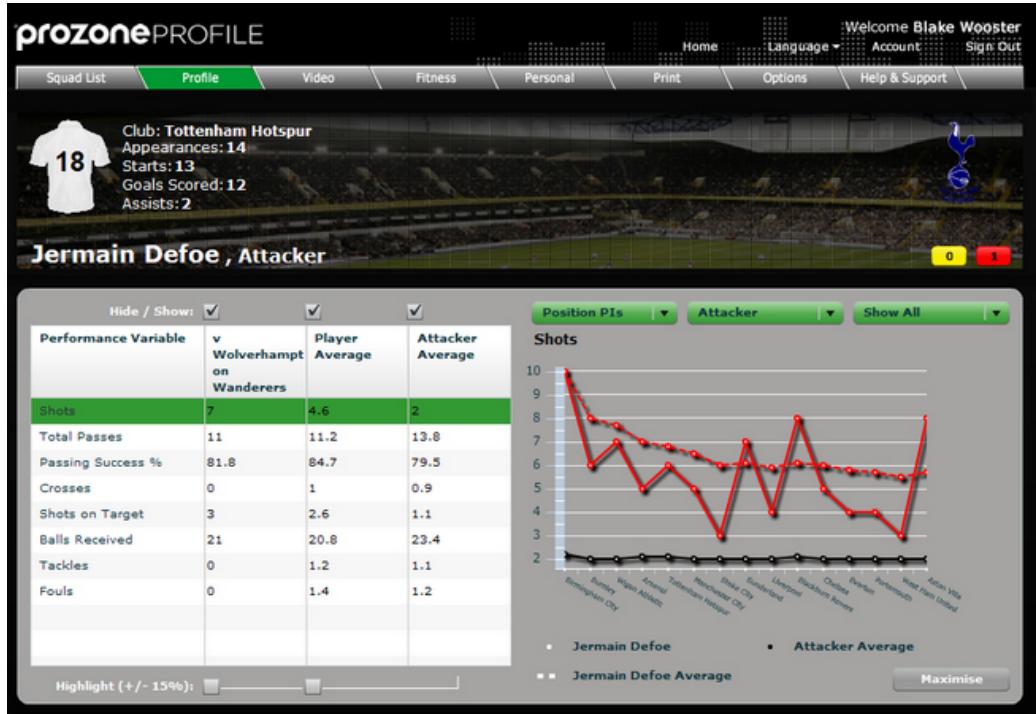


Figure 3.4: The Prozone software - individual player analysis gives you statistics of performance over time. Image from [prozonesports.com](http://prozonesports.com)

amount of time on each exercise. You can get 2D animation of matches as they have tracked every players movement.

### 3.3.2 ZXY

ZXY provides you with a 3D graphic interface showed in figure 3.5. 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 3.5: 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. Image from zxy.no

### 3.3.3 FourFourTwo

FourFourTwo is a monthly magazine about football. They also have a web-page which they update daily. Particular they focus on analytic and has a business deal with Opta giving them access to all theirs statistics. Including to own analyses on their page they have their own application Stats Zone where everyone can analyze matches with the same tools as they do self. Figure 3.6 shows a typical graphical illustration done with the Stats Zone application highlighting passes in to attacking third.

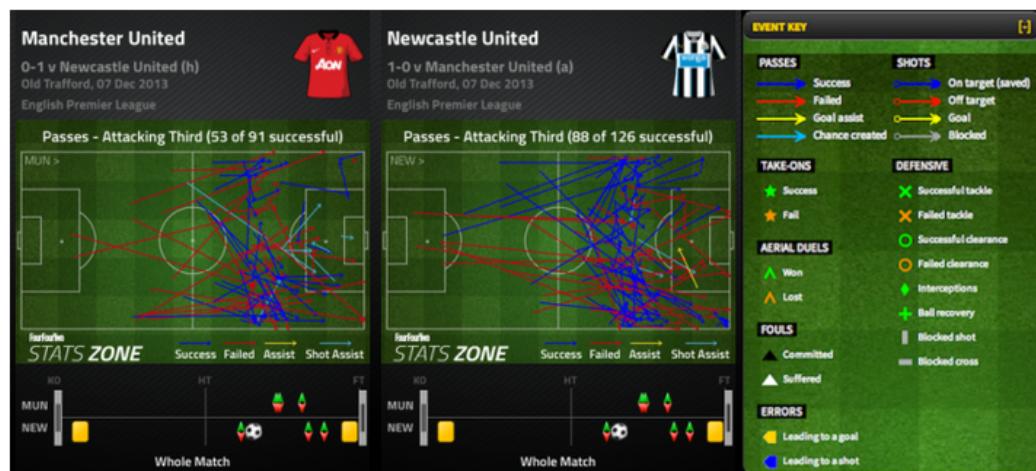


Figure 3.6: Illustration of all passes in to attacking third in the match Manchester United vs Newcastle United. Taken from [fourfourtwo.com](http://fourfourtwo.com)

# Chapter 4

## Requirement Specification

This chapter outlines the requirement specification of the system and the process to get there. The process of gathering the requirements was done in collaboration with Troms IL.

### 4.1 Overview

The requirements evolved during the process of developing the system. Initially a requirement specification was designed from our perspective. We looked at the different anal systems out there and what was in use at Alfheim today. Many systems looks at a single match when analyzing like FourFourTwos application. We are interested to see statistics over time to see if there is patterns of plays that is repeated. The quest of finding which players it is that creates goal scoring chances is the driving force behind the system. It is easy to check who scores the goals and has has the last pass as that kind of statistics is easily found. Players playing in a more defensive role may have the same influence on the results as the strikers.

Rather than going very wide providing all kind of analyses and statistics we narrowed it down to a very concrete system. A system that tries to do for many things may fell between two chairs and at the end doesn't provide anything. When you spend less time on each feature the quality of each feature is most likely reduced.

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. We define key players as players that is often the breakthrough player in the build up to goal attempt by dribbles or passes. Figure 4.2 shows involvements of a typical key player we want to identify.

## 4.2 Capture

A domain model for the captured data is crucial to set as early as possible in the process. A change to the model slows down the development process; captured data has to be re-captured and previous queries on data may fail due to missing fields. The domain model should reflect what we want to get out of the system. It sets the boundary's for which information we can pull from the system afterwards. When defining requirements Troms IL gave us a domain model they have used for a previous analytic project. The project presented a breakdown of all goals for Manchester United in the 2011/2012 season in Barclays Premier League. This domain model shown in figure 4.1 is used as our foundation for our domain model.

Reg_id	Log	Kampnummer	Kamptid	Motstander	Kampested	Mål_number	Tid_mål	Omgang	Målscorer	Scoringsmulighet_pr	Målghenvda	Antall_1_scoringspoi	Corner
	Manchester United	1	8/14/11	West Bromwich	Borte	1	0	13	1 Wayne Rooney	30	Ashley Young	2	Net
Frispark	Skru	Innkast	Straffe	Angrep type	Angrep start	Antall trekk	Målposisjon	Gjennombrudd	Gjennombruddspiller	Innlegg fra	Innlegg til		
Nei	Nei	Nei	Nei	Etablerd spill	Motstanders banehalvdel	14	52b	Pasning Bakrom	Ashley Young	13	\$4		

Figure 4.1: A screen capture showing some data Troms IL captured in a previous analytic project

From this we can define what the visioned system should capture from matches. The visioned system should capture from each match every successful attack that leads to an attempt on goal. Then from each attack you capture where the attack started, every pass including from/to zones, 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 should be stored with a breakthrough player and what type of breakthrough it was. Figure 4.2 illustrates the different breakthroughs we want to capture.

The visioned system needs a data store. The data should be stored in database that has a rich query language for being able to support the large range of queries on the data. Good support for aggregate functions is crucial. The data will consist of text and integers.



Figure 4.2: Illustrations of different breakthroughs we want to register

Additionally an interface for input will be required. This interface should ensure that the input data is correct. For example when defining what type of attack the attack is, only the predefined options should be valid as input.

Capturing every pass in successful attacks requires a positioning system. Here we suggest a more coarse grained dividing of the pitch than x,y co-ordinates like many systems use. We want to look at areas on the pitch. The process of defining how to divide the pitch into zones took several rounds of discussion leading to several types. These are shown in figure 4.4.

### 4.3 Present

The presentation of the data is an important aspect of the system. This will be where the end-users will spend their time. The presentation of data needs to be as simple as possible to understand. The coaching staff is no technical experts and the system is aimed for them to use. Therefore the system will need to present statistics and other valuable information in clean and understandable way. In the end it is the 11 players the coach selects that perform all the actions, but the coach sets the style of play and boundaries for players. The goal is to give the coach the opportunity to use the system, using both analytic illustrations and statistics, so he can better reach out to his players with his ideas. The better a coach can sell his ideas to the players the more they will believe in the philosophy, style of play and follow the game plan in games.

We purpose a web interface here as it is accessible from many devices as long as you have an Internet connection and a up to date web browser.

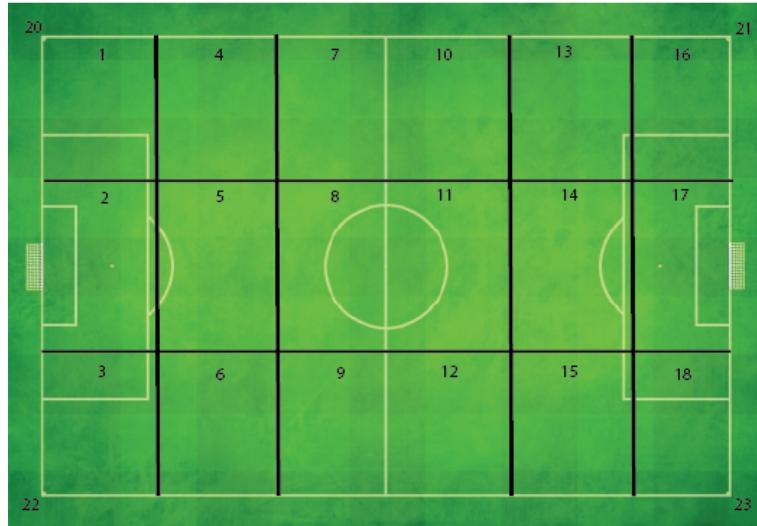


Figure 4.3: Dividing of pitch - the first suggestion had 18 zones

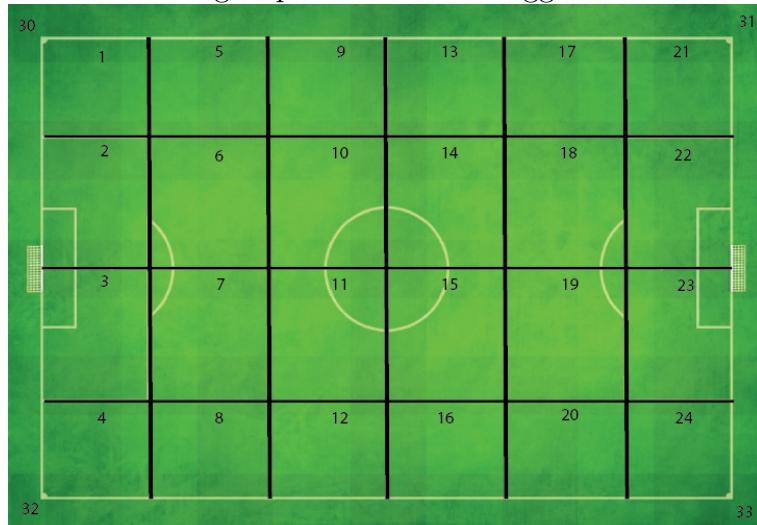


Figure 4.4: Dividing of pitch - the second suggestion had 24 zones

## 4.4 Summary

To summarize we need storage, back-end and an interface for capturing and presenting analytic information. Figure 4.5 shows the conceptual architecture.

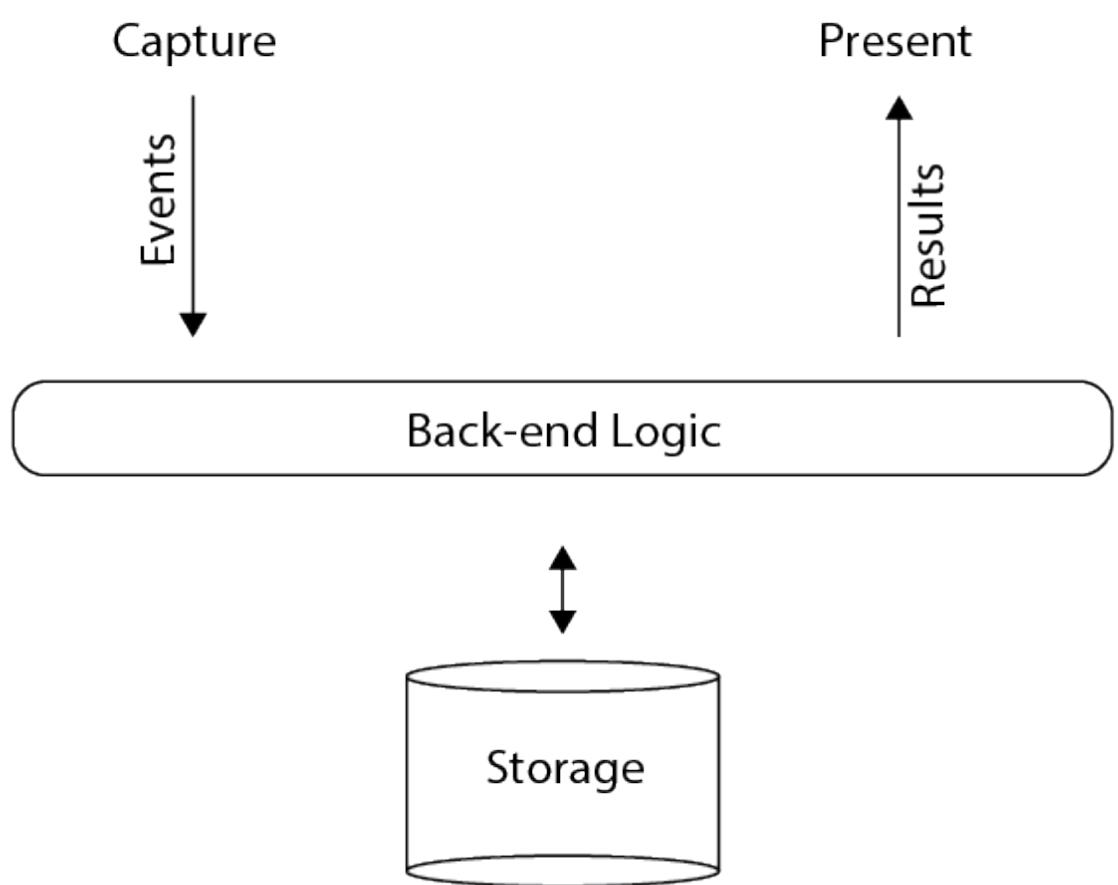


Figure 4.5: Conceptual architecture



# Chapter 5

## Soccer Analytic toolkit

### 5.1 System Architecture

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.

#### 5.1.0.1 Interface

The Soccer Analytic tool has one user interface - the web browser interface. Both the input and the analytic interface can be reached from this single web browser interface.

#### 5.1.0.2 Back-end

The back-end is the middle layer between the client and the storage. Its main task is to serve static files to the clients, handle data insertions or handling web request by mapping them to database operations. New data insertions will possibly be inserted into several database indexes. The back-end will then ensure that all indexes are updated before returning success to the client.

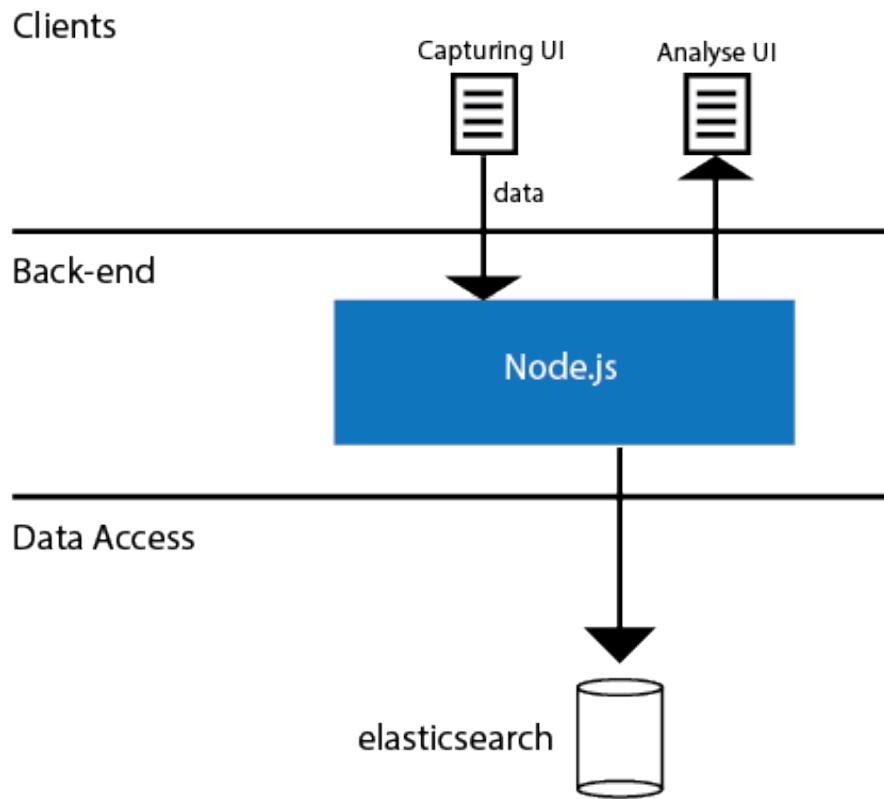


Figure 5.1: Overall architecture of the system

#### 5.1.0.3 Storage

The storage layers task is to persist data and handle search queries on the data. It consists of several indexes that each stores a part of the domain model.

#### 5.1.1 Domain model

The domain model is based around matches. All attacks are wrapped into a match root. Attacks can be seen as subdocuments of the match document. Inside the attacks all passes lies with other information. This gives an easy way of understanding how all the data is related together. Below is a complete example of how it all is structured. The number of attacks and passes is stripped down to one in the example.

```

{
  "hometeam" : "Trøms",
  "awayteam" : "Rosenborg",
  "score" : "1-0",
  "date" : '2013-15-09',
  "attacks": [
    {
      "time": 4,
      "touch": 1,
      "team": "Trøms",
      "breakthrough": "None",
      "breakthroughPlayer": "None",
      "typeOfAttack": "Dribball",
      "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"
      }
    },
  ]
}

```

Figure 5.2 shows the final version of the pitch dividing with a illustration of how an attack is captured. Player 4 plays to player 10 from zone 13 to 14. Player 10 transports the ball until he passes to player 15 from zone 18 to 22. Player 15 tries a finish from zone 22 inside the penalty area. The reason for going for more zones in the penalty area was to be able to see more specific where attacks are finished.

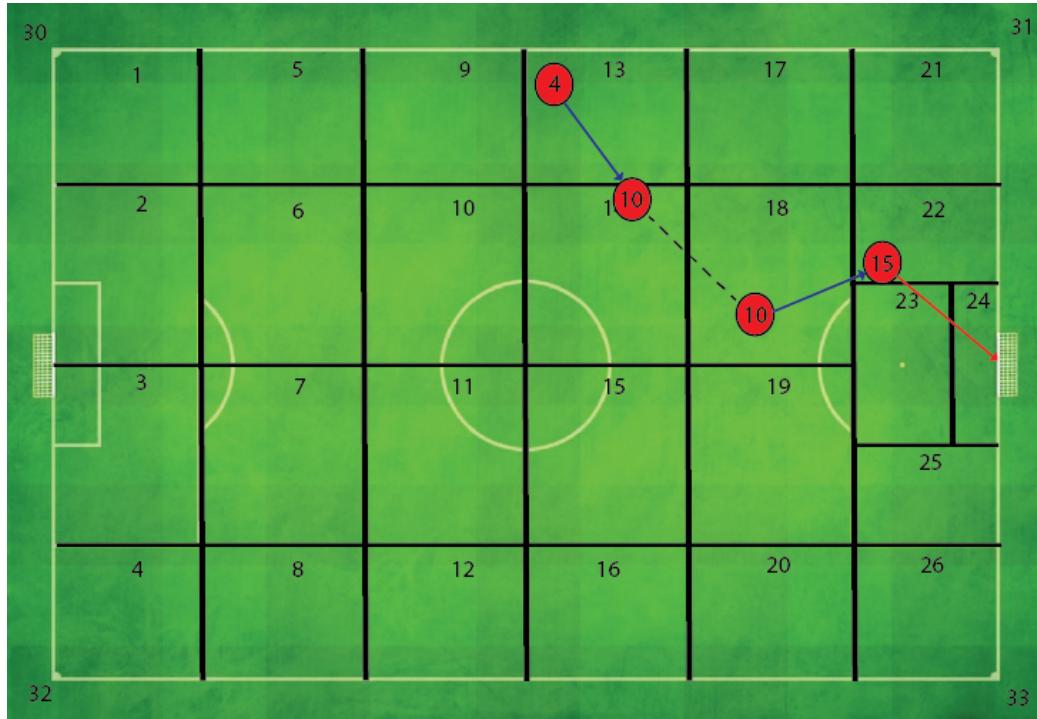


Figure 5.2: Final pitch dividing. More zones have been added to the penalty box area than the initially design.

## 5.2 Implementation details

Software Stack	
<b>Web server</b>	Node.js
<b>Database</b>	Elasticsearch
<b>Web Interface</b>	HTML5, CSS3
<b>Client Side Technology</b>	Backbone, Bootstrap, jQuery, Mustache, Highcharts, Lightbox, async

Figure 5.3: Software stack

### 5.2.1 Storage

The data storage is an elasticsearch database<sup>1</sup>. Elasticsearch is document oriented, schema less and works well with JavaScript Object Notation (JSON)<sup>2</sup>.

<sup>1</sup><http://www.elasticsearch.org/>

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, and make it available for search. Elasticsearch takes advantages of embedded documents meaning we can store related data together. An attack is usually made up of several passes these can be stored as an embedded document in the attack document. Then all passes can be retrieved in one query when fetching an attack.

The main reason for using Elasticsearch is its search capability. In the starting phase of the project MongoDB <sup>3</sup>was the storage engine. However as some aggregation queries was hard to figure out how to do a change to Elasticsearch was made. It should be noted that MongoDB supports map reduce operations. With some extra work the queries causing problems could possibly have been done with MongoDB. However, with Elasticsearch you can in a single, simple to write query get counted how many passes all players for a team has played and received, the number of times all players has been the breakthrough-player in an attack, count type of attacks, count the most used zones for passing and finishing and so on. This makes it very easy to do queries for analyses on teams and players.

### 5.2.1.1 Indexes

Indexes in much like tables in a relational database in the way that they are a container for data. An index stores documents, which is a bunch of key/value pairs, like JSON. You can let Elasticsearch automatic analyze the field data or you can use mappings. For supporting querying on players and other fields where the value is more than one word, you have to tell Elasticsearch that it is a multi field. For example searching for player name "Stefan Johansen" when the field is not set as multi field will give you two results and not one, as you would expect.

In our project we have 5 different indexes. You have Team index, which holds all the teams. Player index which holds all the players. Match index holds and stores all data from a match including all attacks. Attack index stores information about attacks. Last you have the Pass index, which stores passes. Listing 5.1.1 shows an example of data stored in the match, attack and pass indexes. As may be noted there is data redundancy. You can argue that storage has become so cheap and if you can use a little bit extra space

---

<sup>3</sup><http://www.json.org/>

<sup>3</sup><http://www.mongodb.org/>

to gain performance you would do it. In this case it was done to be able to fully support aggregation functions on subdocuments.

### 5.2.2 Back-end

The back end is the middle-ware between the clients and the data layer. It exposes a Representational state transfer (REST)<sup>4</sup> interface over Hypertext Transfer Protocol (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. The server will respond with HTTP status code 201 if all goes well or 400 on an error. The server uses HTTP code actively to tell the client the result of requests.

In principle, since the data input is generated in the web browser (with JavaScript), it could have been inserted right away into the database without going through an extra middleware. However, this limits us, as we can't combine multiple queries by going through the back-end. Cleaning of data before serving to the client would neither be possible. Normally you also do validation of the data at the back-end before inserting into your database.

#### 5.2.2.1 API

The Application Programming Interface (API) of the web server is listed in figure 5.4. As mentioned the API follows the REST principles.

### 5.2.3 Front-end

Front end is consisting of a single page JavaScript application using Backbone.js<sup>5</sup> as under-supporting framework. Backbone uses a Model View Presenter (MVP) model to structure the code. With Backbone your views will update automatically when data changes. In the following sections the architecture of the client and how different concepts are used will be described.

---

<sup>4</sup><http://www.ibm.com/developerworks/webservices/library/ws-restful/>

<sup>5</sup><http://backbonejs.org/>

Web server API	
<b>GET /matches</b>	Get all matches
<b>GET /match/:id</b>	Get specific match
<b>POST /match</b>	Post new match
<b>GET /teams</b>	Get all teams
<b>GET /team/:name</b>	Get team statistics
<b>GET /team/:name/finalthird</b>	Get passes into final third of the pitch
<b>POST /team</b>	Post new team
<b>GET /player/:id</b>	Get player statistics
<b>GET /player/:id/breakthroughs</b>	Get breakthroughs for player
<b>GET /player/:id/finalthird</b>	Get passes into final third of the pitch
<b>POST /player</b>	Post new player
<b>POST /attack</b>	Post new attack
<b>POST /pass</b>	Post new pass

Figure 5.4: Overview of the web servers API

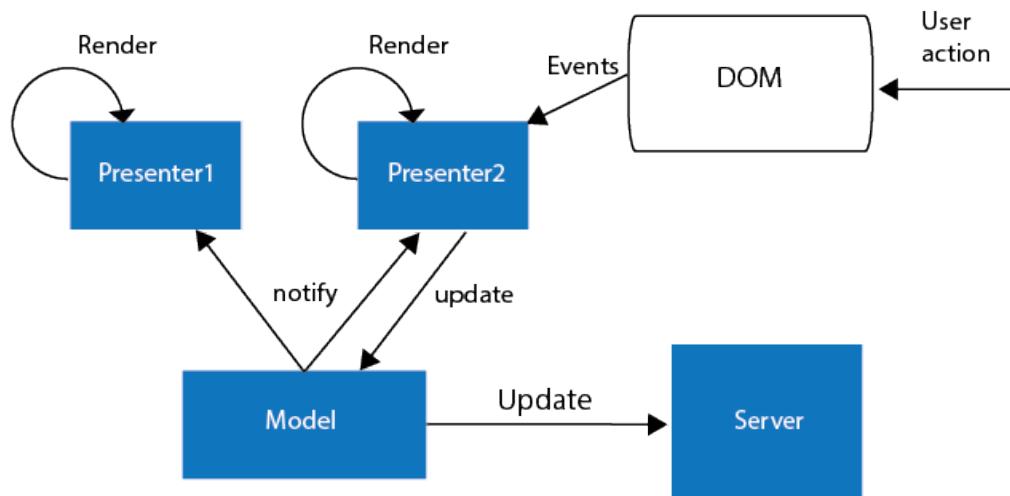


Figure 5.5: Architecture of the client side

### 5.2.3.1 Models

Data is represented as models in backbone. The model has mainly two responsibilities. First is whenever an update on a models data occurs the model

notifies the presenters that have subscribed for update events for that particular model. The second is that models are responsible for Asynchronous JavaScript and XML (AJAX) communication with the back-end. An example is when a user registers a new attack for a match. He fills out a form and press the submit button. Then a new Attack Model is created. Calling save on the instance of the model will send an AJAX post request to the server with the models data in the HTTP body.

Similar to an Attack Model we have Player Model that handles everything around players. When you click yourself into a players profile the model will fetch statistics from the back-end, notify the presenter that the data is ready, and the HyperText Markup Language (HTML) will be rendered in the web browser.

There is also a Match Model for fetching and registration of matches, Team Model for viewing statistics, and a Pass Model for saving passes. They work it the same way as the models described above using AJAX to communicate with the back-end and notification presenters on data changes.

#### 5.2.3.2 Presenters

Presenters are the clue between the Model and the View (web page). Presenters can bind own functions to respond to models events. For example on data change event from a model the presenter will call its render function and the User Interface (UI) will updated. Each presenter has a own HTML div tag which he renders to. Including to subscribing to events on models the presenter can subscribe for view events that happens in the Document Object Model (DOM), events that the user triggers. Following up the example in the previous section on an attack registration. When a user press the submit button the presenter will be notified. He will create a model for the attack and calling the save method on the instance object.

#### 5.2.3.3 Views

For a analytic toolkit to be useful a good presentation is critical. Here several helper libraries are used to present the data. Highcharts.js<sup>6</sup>is a 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 plots it in the same graph. In figure 5.6 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 is shown.

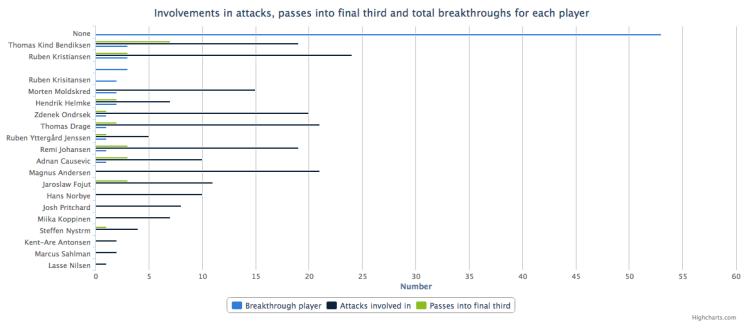


Figure 5.6: Shows how the passing statistic is illustrated on the client by using Highcharts.js

Positional data is created using the new feature of HTML5, canvas element. It lets you draw graphics on the fly in the web page. In this system it is used to create an element that symbols the different zones in our domain model. In the Team Model you have all zones with a number that symbols shots taken from that zone. This is then plotted into the respective zones as figure 5.7 is showing.

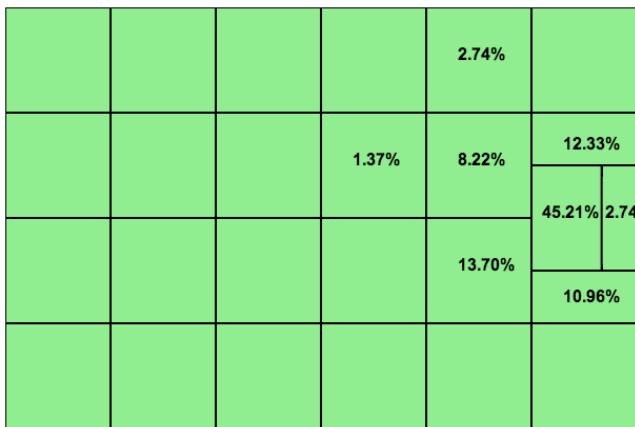


Figure 5.7: Illustrations of which zones the team has finished off their attacks from, with percent. Team - Troms IL)

Backbone comes with a library Underscore.js<sup>7</sup>that makes creating HTML pages with dynamic content easily. When you are rendering new content

<sup>6</sup><http://www.highcharts.com/>

on the site you can insert data retrieved from a model dynamically into the HTML. In the example below the input is a an array of objects where each element in the array contains a key value pair 'name' : value. The HTML output of this will be a list of names.

```
{#{players}}  
    <li>{{name}}</li>  
{/players}
```

#### 5.2.3.4 Router

A component not mentioned before that lies on the client is the router. The router is the glue that binds all the other modules together. It handles the navigation between pages in the application. When users navigate on the page by clicking on links or buttons in a traditional web page the back-end serves a new HTML page. Here, a router module handles this click event. The router model examines the URL request and routes the request to the view responsible for that specific route. The view will then call fetch functions on the model to get data from the back-end and render the HTML.

#### 5.2.4 Getting players and teams into the database

Getting squads in a useful format like eXtensible Markup Language (XML) or JSON was harder than expected. On the Norwegian football alliance's website you could download the squads for the current season, but only in PDF format. Current squads is fetched from altomfotball.no, a website by the Norwegian broadcasting company TV2. The fetching process is an own python script meant to run only once to set up the database. For each team the script basically reads the HTML document with all players listed, parse out players name, and sends in to the web server.

#### 5.2.5 Security

Security is not taken into concern. This means anyone getting into the page can post new match data and add attacks. This could have been fixed by requiring a login before getting access to the site.

---

<sup>7</sup><http://underscorejs.org/>

# Chapter 6

## Demo

### 6.1 Interfaces

#### 6.1.1 Mockup

Figure 6.1 shows a mockup of the team analysis page that was created before starting developing. There are two main sections in the mockup: team statistics and key players.

#### 6.1.2 Implemented interfaces

##### 6.1.2.1 Home page

The first page you are prompted with is the listing of all matches registered in the database as figure 6.2. A click on match gives you details about that match and prompts you an interface for capturing new attacks if requested. Every field has to be submitted with a correct input value.

The whole website follows a theme throughout the pages. This theme is the standard theme in the CSS and JavaScript library Bootstrap<sup>1</sup>. Bootstrap makes the website by default responsive. This means the content on the site is automatically re-sized out from the size of your browser window.

---

<sup>1</sup><http://getbootstrap.com/>

### **6.1.2.2 Registering attacks**

You register a new attack by pressing the New attack button. Here you fill in the result and date of the match. When you submit the match you will be sent back to the home page.

### **6.1.2.3 Match view**

Clicking on a match gives you overview of all attacks registered for that particular game shown in figure 6.3). This view is only meant for quality checking the data.

From the match page you can add new attack attempts, as figure 6.4 shows. This will prompt you with a fill in form for the attack. Attack start, attack end, breakthrough player, time and team needs to be filled out. The passes in the attack are added by pressing new pass. This will add a new pass to the form.

### **6.1.2.4 Team view**

Then you have the team selection page shown in figure 6.5). This is a simple layout listing all the teams in the Norwegian premier league. From here you select the team you want to analyze.

Then there is the main view used for analyzing a team, showed in figure 6.6, figure 6.7 and figure 6.8. The page design is not exactly the same as the mockup. This comes from a various things. A description of different breakthroughs has been added to enlighten users of the system. Users of the system may or may not have been involved in the process of capturing the data and therefor-extra information is needed to clarify concepts. In the mockup key players where highlighted. In the final view various statistics is presented to the user and he can then click on players he find interesting to know more about as shown in figure ???. The whole page is divided into 3 sections; Key aspects, offensive play, defensive play.

Clicking on a player brings you to the player view shown in figure 6.10 and figure 6.11. Here individual statistics is highlighted.

## 6.2 Capturing process

As mentioned, the process of capturing data is manually. In the beginning it took up to 1 hour to capture all attacks for a match. When you get used to the interface and is able to quickly identify players the time used went down 15-20 minutes. Where most of the time went was getting the players id by looking up in the database manually.

The process of capturing has been the following:

1. Download the match video.
2. Find the match in the VGLive.no archives. VGLive.no is used to quickly find all attacks ending with a finish
3. Capture all the attack one by one via the implemented interface mentioned earlier.

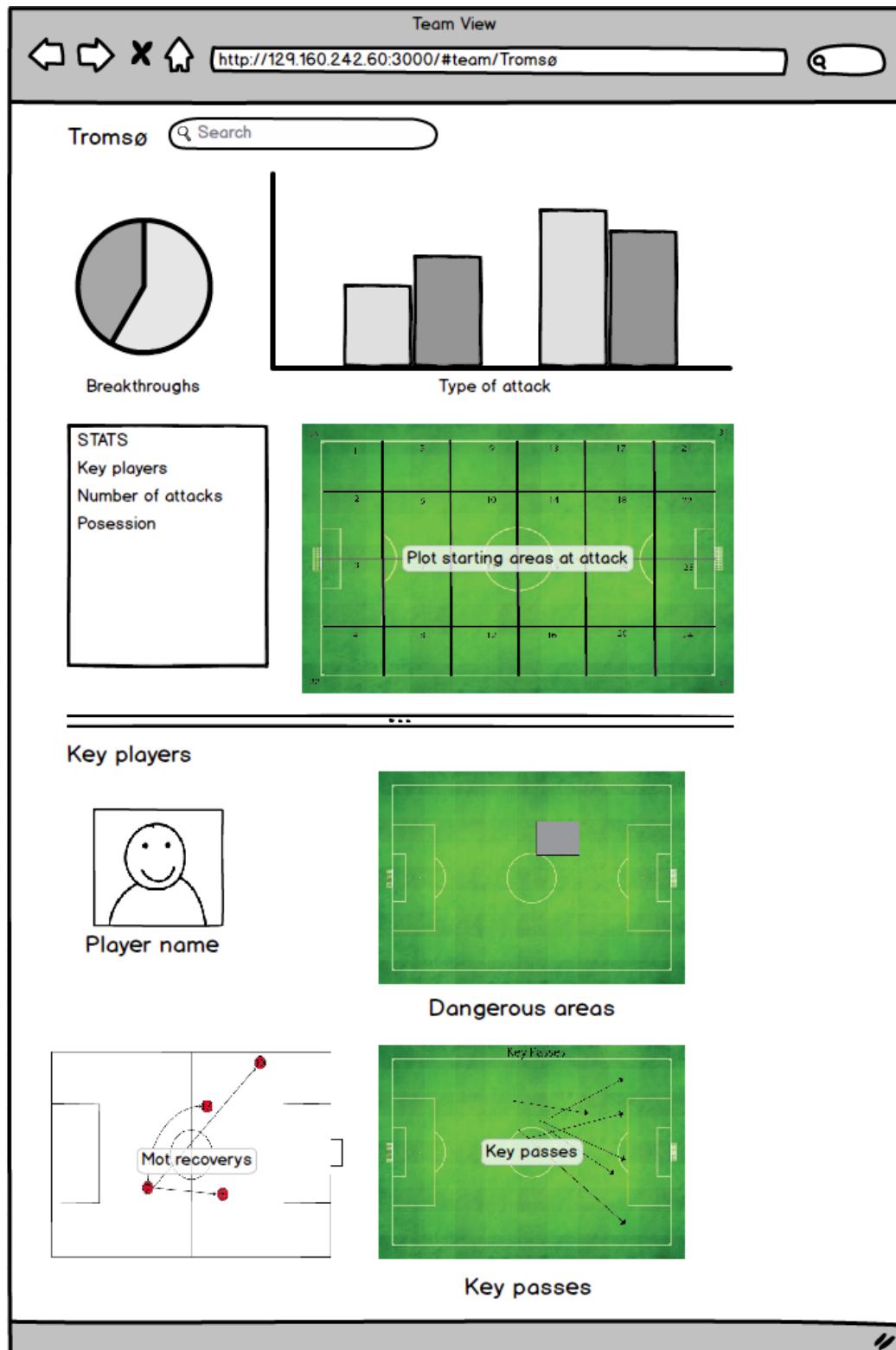


Figure 6.1: A mockup of the main analytic page  
34

<a href="#">Home</a>	<a href="#">Teams</a>	<input type="text" value="Search"/>
<b>Latest matches</b>		
<a href="#">New match</a>		
<b>Tromsø - Start, 2013-09-29</b>		
<ul style="list-style-type: none"> <li>Score: 2-3</li> <li>Registered attacks: 6</li> </ul>		
<b>Tromsø - Ålesund, 2013-08-18</b>		
<ul style="list-style-type: none"> <li>Score: 1-2</li> <li>Registered attacks: 12</li> </ul>		
<b>Strømsgodset - Start, 2013-10-07</b>		
<ul style="list-style-type: none"> <li>Score: 1-0</li> <li>Registered attacks: 10</li> </ul>		
<b>Tromsø - Viking, 2013-10-19</b>		
<ul style="list-style-type: none"> <li>Score: 4-3</li> <li>Registered attacks: 14</li> </ul>		
<b>Strømsgodset - Tromsø, 2013-06-29</b>		
<ul style="list-style-type: none"> <li>Score: 3-1</li> <li>Registered attacks: 5</li> </ul>		
<b>Sarpsborg - Tromsø, 2013-09-22</b>		
<ul style="list-style-type: none"> <li>Score: 0-3</li> <li>Registered attacks: 6</li> </ul>		

Figure 6.2: All matches registered in the database are listed on this page

## Tromsø - Start

New attack

- Chances registered: 6:
  - 40. min - Gjenvinning kort angrep (Start) ✓
    - Passes: 2
    - Touches: 4
    - Breakthrough player:
      - Håkon Opdal, Pasning bakrom
    - Attack start:
      - Gjenvinning, 341 zone 5
    - Finish: SHOTGOAL from zone 17
  - 48. min - Dødball (Start) ✓
    - Passes: 1
    - Touches: 3
    - Breakthrough player:
      - None, None
    - Attack start:
      - Corner, 349 zone 21
    - Finish: SHOTGOAL from zone 17
  - 62. min - Gjenvinning kort angrep (Start) ✓
    - Passes: 1
    - Touches: 3
    - Breakthrough player:
      - Espen Hoff, 1vs1 Mellomrom
    - Attack start:
      - Gjenvinning, 347 zone 11
    - Finish: SHOTGOAL from zone 17
  - 67. min - Gjenvinning langt angrep (Tromsø) ✓
    - Passes: 2
    - Touches: 7
    - Breakthrough player:
      - None, None
    - Attack start:
      - Gjenvinning, 402 zone 11
    - Finish: SHOTGOAL from zone 17
  - 71. min - Gjenvinning langt angrep (Tromsø) ✓
    - Passes: 4
    - Touches: 7
    - Breakthrough player:
      - None, None
    - Attack start:
      - Gjenvinning, 393 zone 10
    - Finish: SHOTGOAL from zone 17

Figure 6.3: Interface listing up all attacks for a match

**Attack attempt** [Collapse this section](#)

[Remove attack](#)

**General stuff**

\$ Time of attack

\$ Team

None

\$ Breakthrough Player

Etablert spill

**Attack start**

\$ Zone

Gjenvinning

\$ Player ID

**Register pass**

[Remove pass](#)

\$ Pass from (player id)

From zone

To player (player id)

To zone

Action (CROSS, PASS, KEYPASS, LONGBALL)

[New pass](#)

**Attack finish**

\$ Player ID

\$ Zone

ShotMISS

[Submit](#)

Figure 6.4: Interface for registering an attack

## All Teams

Viking
Start
Hønefoss
Rosenborg
Molde
Brann
Sandnes Ulf
Aalesund
Haugesund
Vålerenga
Strømsgodset
Sarpsborg 08
Lillestrøm
Odd
Sogndal
Tromsø

Figure 6.5: Interface listing all teams

## Tromsø

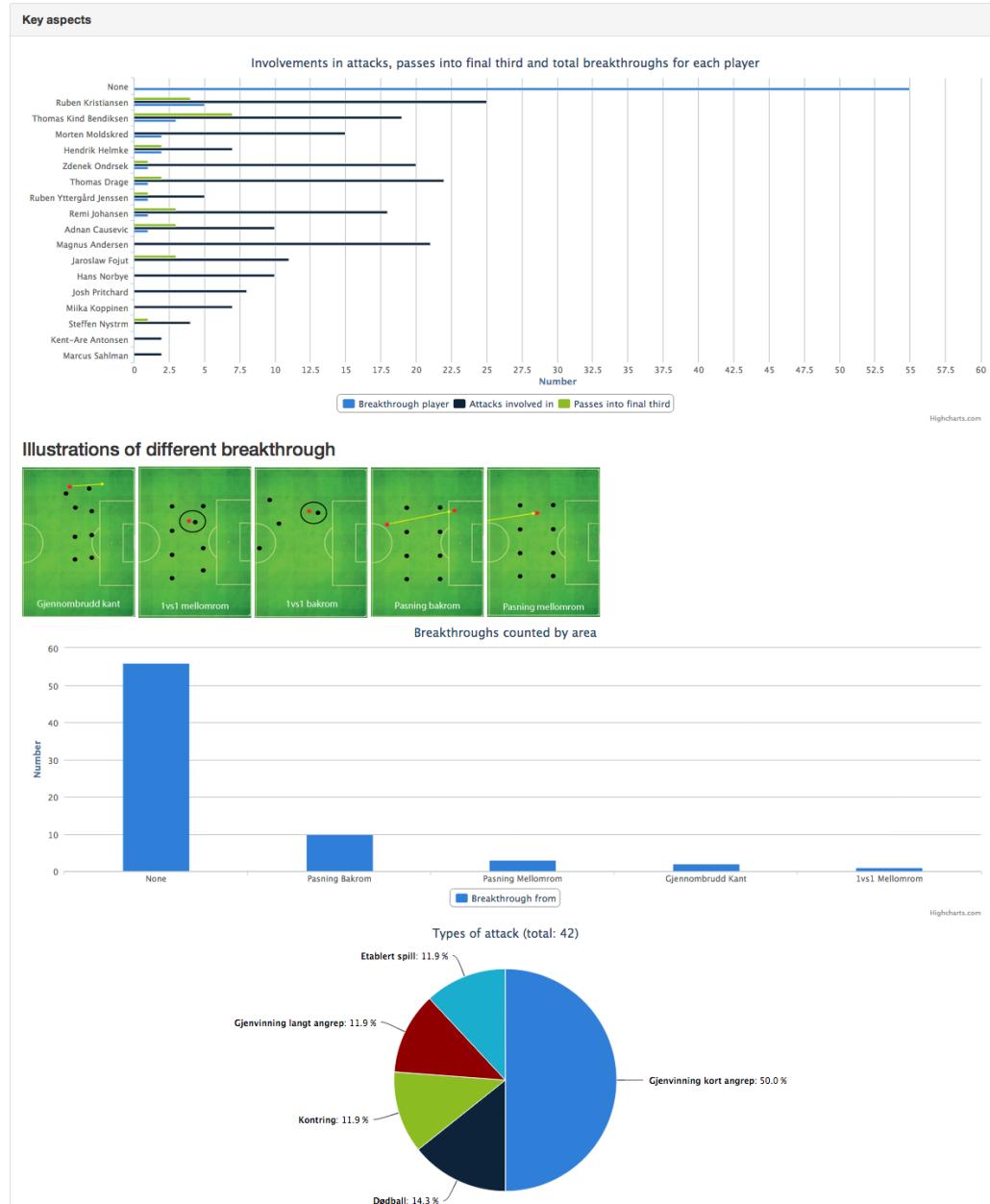


Figure 6.6: Main interface for information about opponents - key aspects

## Offensive play - attacking from left to right

### Attacking zones: Which zones is used most in attacks?

Statistic is generated by counting all passes from a zone

0.60%	1.20%	3.01%	6.63%	4.52%	2.41%	0.30
0.90%	0.60%	9.64%	9.94%	6.93%	1.81%	
0.90%	1.81%	6.02%	7.83%	7.83%	4.52% 3.61	2.41%
0.30%	0.30%	1.81%	6.63%	5.42%	1.51%	0.60

### Finishing: Where has the attacks been finished from?

Statistic is generated by counting all finishes from a zone

				7.69%	10.26%	
				12.82%	46.15% 15.3	5.13%

Figure 6.7: Main interface for information about opponents - offensive play

## Defensive play - attacking from left to right

Attack start: Where is the attack started from?

Statistic is generated by counting all starting zone for each attack

	<b>2.56%</b>	7.69%	5.13%	5.13%	<b>2.56</b>
<b>2.56%</b>	<b>5.13%</b>		7.69%	<b>2.56%</b>	<b>2.56%</b>
<b>2.56%</b>	<b>2.56%</b>	<b>5.13%</b>	<b>2.56%</b>	<b>10.26%</b>	<b>2.56%</b>
	<b>2.56%</b>		7.69%	<b>10.26%</b>	<b>5.13</b>

Figure 6.8: Main interface for information about opponents - defensive play

## Players

- [Marcus Sahlman](#)
- [Fredrik Bakkelund](#)
- [Benny Lekstrm](#)
- [Jaroslaw Fojut](#)
- [Ruben Kristiansen](#)
- [Adnan Causevic](#)
- [Mika Koppinen](#)
- [Hans Norbye](#)
- [Mathias Johnsen](#)
- [Kent-Are Antonsen](#)
- [Jonas Hylo Fundingsrud](#)
- [Lasse Nilsen](#)
- [Thomas Kind Bendiksen](#)
- [Thomas Drage](#)
- [Magnus Andersen](#)
- [Lars-Gunnar Johnsen](#)
- [Remi Johansen](#)
- [Josh Pritchard](#)
- [William Johan Frantzen](#)
- [Hamza Zakari](#)
- [Hendrik Helmke](#)
- [Morten Moldskred](#)
- [Steffen Nystrm](#)
- [Zdenek Ondrasek](#)
- [Runar Espejord](#)
- [Ruben Yttergård Jenssen](#)

Figure 6.9: Main interface for information about opponents all players

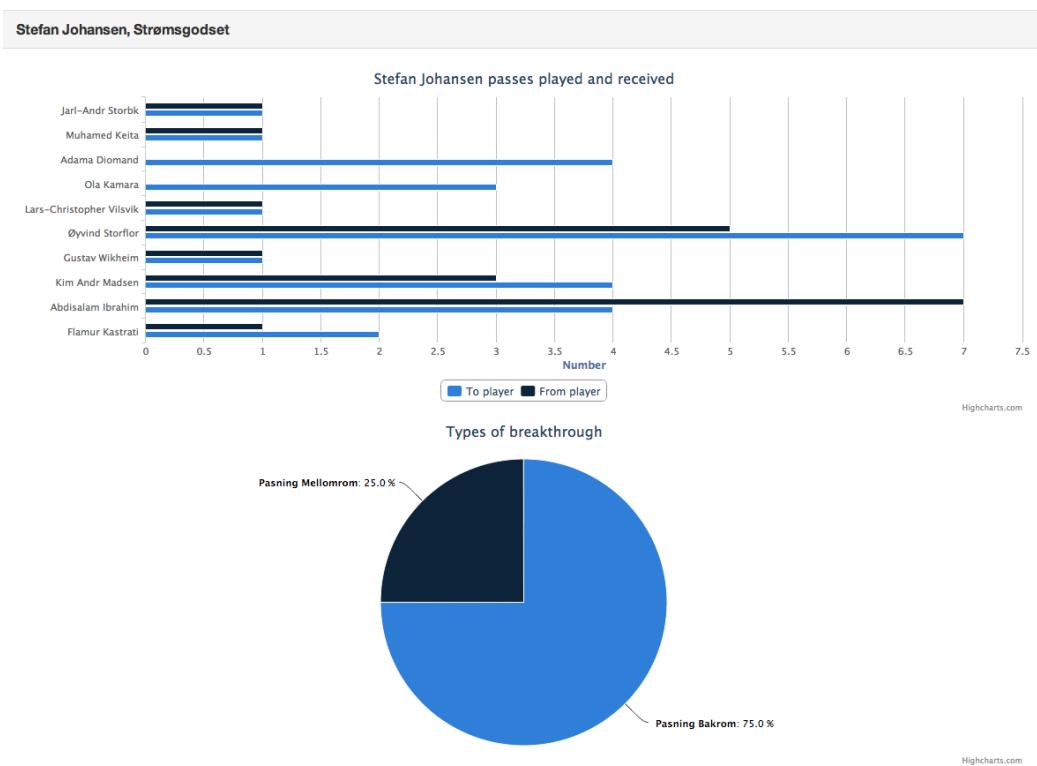


Figure 6.10: Player view highlighting individual statistics

**Zones played the ball from**

		3.03%	6.06%	3.03%	
3.03%		9.09%	12.12%	6.06%	
	3.03%	9.09%	27.27%	3.03%	
				6.06%	3.03%
					6.06

**Zones played the ball to**

		4.00%		4.00%	
4.00%		4.00%	12.00%	12.00%	
	4.00%	8.00%	36.00%		4.00%
				4.00%	4.00%

Figure 6.11: Player view highlighting individual statistics

# Chapter 7

## Evaluation and Results

This chapter presents methods used to evaluate the system and the results collected evaluating the system.

### 7.1 Methods

#### 7.1.1 User Survey

The system is measured by doing user surveys on the end users. In our user survey we have coaches rating how much they agree with a statement on a Likert-scale. The statements compare the system against other systems in use at Alfheim today. The Likert-scale chosen is a 5 point scale from *strongly disagree* to *strongly agree*.<sup>1</sup>. In short it will let each individual to note how much they disagree or agree with a particular statement.

Including to this external people not in the Troms IL system. This includes Ruben Yttergrd Jenssen playing in the national soccer team for Norway and FC Kaiserslautern in 2. Bundesliga, and Lars Tjørns, a previous coach and now football expert in Canal+ also writing for 100 fotball.

---

<sup>1</sup><http://www.simplypsychology.org/likert-scale.html>

## 7.2 Experiments and Results

### 7.2.1 Test data and Assessors

In the tests Troms IL academy coaches has been evaluating the system. In the evaluation phase the database had been populated with data from Troms IL and Strmsgodset Toppfotball matches. Only attacks from these two teams have been used in the evaluating process. A total of 34 attacks have been captured for Strmsgodset over 5 matches. For Troms a total of 42 attacks have been captured over 9 matches. Figure 7.1 lists all matches. Some matches include data from other teams than Troms and Strmsgodset. They have not been taken into consideration in the evaluating process.

Match	Attacks registered
<b>Viking - Tromsø, 2013-05-26</b>	8
<b>Strømsgodset - Tromsø, 2013-06-29</b>	5
<b>Molde - Tromsø, 2013-06-10</b>	14
<b>Strømsgodset - Vålerenga, 2013-05-10</b>	8
<b>Strømsgodset - Start, 2013-10-07</b>	8
<b>Tromsø - Vålerenga, 2013-08-04</b>	15
<b>Tromsø - Start, 2013-09-29</b>	6
<b>Tromsø - Viking, 2013-10-19</b>	14
<b>Tromsø - Ålesund, 2013-08-18</b>	12
<b>Tromsø - Strømsgodset, 2013-11-03</b>	15
<b>Sarpsborg - Tromsø</b>	5
<b>Sogndal - Strømsgodset, 2013-09-27</b>	13

Figure 7.1: Matches that have been captured and persisted into the database

### 7.2.2 SAT as a tool for opponent analytic

In this User Survey the assessors where asked how SAT is as a complementary opponent analytic tool. In the requirement process we specified that the system should to be a complementing tool and not a direct replacement of the systems in use. The results shows that most of the assessors values what the system provides.

Assessors not in the Troms IL system do not necessarily use or have experience with an existing tool. Therefor the statement was re-phrased. The

assessors were asked how SAT is as an opponent analytic tool, a stand-alone product. The results shows

From these results we can conclude that the end-users view the Soccer Analytic Tool as a good system to complement with for opponent analytic.

### **7.2.3 SAT as a tool for identifying key players**

In this User Survey the assessors where asked how SAT is as a tool for identifying key players in a soccer team. This was one of the main goals of the system. We can see from the results of the User Survey that the system identifies key players exceptional well. All votes for strongly agree.

### **7.2.4 UI-performance**

## **7.3 Discussion**

### **7.3.1 Input**

Perhaps the biggest limitation of the system is that it requires manual work to be functional. The system requires a lot of manually work to be operational as an opponent analytic tool. Up to one hour is the normal time spent capturing all data for a match with the current capture interface. On average, teams across the top four top leagues (La Liga, Premier League, Serie A and Bundesliga) took about 13 shots per match, measured in the 2010/11 season [6]. This means you will have possible have 26 attacks to capture (not all shots can be seen as an attempt on goal).

As mentioned in section 6.2 you have to store the players by their ID. IDs are only found in the database. Instead of this a player selector interface could have been developed letting you just click on a image of the player involved. We suggest you can save up to 20 minutes by having this feature added.

There is minimal quality checking of the input data except from the match view page that lists every attack captured, meant for external operators to verify the data. Other than that you have to trust the operator that is capturing match data. The input is to some degree subjective for some data like identifying breakthroughs. In some situations one operator may say that

it was a breakthrough and another wouldn't agree. This may not be the biggest problem if you set some rules to follow for the operators.

### **7.3.2 Interfaces**

One comment of end users was about the graphical components illustrating statistics on the client. Improving the graphical components would increase the overall experience of the system. This has not been one of the main focuses during the development of the system.

### **7.3.3 Scalability**

# Chapter 8

## Conclusion

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

In this project, develops and evaluates a system for capturing, persisting and presenting information in the field of soccer opponent analysis. Particularly we have focused on identifying key players in the opponent team

*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 key players in a team. A user interface component providing the core information about the opponent should also be developed.*

In the requirement specification we stated what we wanted from the system; a system that identifies key players in the offensive play of a soccer team. We also wanted to be able to see where on the field the key players contribute from.

### 8.0.4 Concluding Remarks

## 8.1 Future Work



# References

- [1] Valter Di Salvo Adam Collins Barry McNeill Marco Cardinale. Validation of prozone ®: A new video-based performance analysis system. *mordi*, 2006.
- [2] 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.
- [3] Håvard D. Johansen Svein Arne Pettersen Pål Haloversen and Dag Johansen. Combining video and player telemetry for evidence-based decisions in soccer. *Regional Centre for Sport, Exercise and Health - North*, 2013.
- [4] Martin Hardy. How the science of opta and prozone’s statistics are changing the premier league, Nov 2011.
- [5] Jonathan Howard. Sports science and big data opportunities, October 2013.
- [6] Anderson Sally. Shots and goals in the big 4: Offensive performance across leagues in 2010/11. <http://www.soccerbythenumbers.com/>, aug 2011.
- [7] Jason Turbow. Soccer embraces big data to quantify the beautiful game, June 2012.
- [8] Mark Venables. Sportstech: Football, 2013. [Online; accessed 12-November-2013].