Design of a coaching support system for evaluating sport tactics

Master's Thesis

Arjan Peters

<<Page left blank intentionally>>

Design of a coaching support system for evaluating sport tactics

THESIS

Submitted in the partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Arjan Peters
Born in Tilburg, the Netherlands

Web Information Systems
Department of Software Technology
Faculty, EEMCS, Delft University of Technology
Delft, the Netherlands
eecms.tudelft.nl

ORTEC TSS
Groningenweg 6k
Gouda, the Netherlands
tss.ortec.nl

Design of a coaching support system for evaluating sport tactics

Author: Arjan Peters
Student ID: 1174711
Email: a.peters@student.TUDelft.NL

Abstract

Coaches are using video analysis to evaluate the performance of a team in field hockey. Events with the ball are tagged and saved in a match annotation file, which is used to lookup video fragments. The match annotation enables a coach to lookup and to evaluate single moments of the match but doesn't provide a coach insight in the tactic. Tactic doesn't relate to single event but to a sequence of events. Currently it is not possible to evaluate the tactic.

Because the match annotation contains information about the events, which includes the time and the location it is spatial-temporal dataset. To lookup a tactical pattern a coach it is required to express a tactical pattern on the spatial-temporal dataset. To enable a coach to incorporate his view of the game the flexibility is important.

In this thesis work a system is designed which enables a coach to evaluate tactic by enabling him to express a tactical pattern and by visualizing the results. A prototype of the system is implemented to evaluate the design of the system. Coaches active at the top national and international level provided input for the system and participated in the evaluation of the prototype.

The designed system provides a useful method to evaluate tactics by a coach. The system enables the expression of a tactical pattern by a coach in a flexible manner. This thesis presents the requirements for a system to analyze tactics. It also presents an approach to model and express a tactical pattern using a domain specific language. To create a tactical pattern a visual query interface is presented. The visual interface also offers the possibility to interpret the result of a query.

Keywords: domain specific language, spatial-temporal querying, field hockey, visual query interface

Graduation Committee:
Prof. dr. ir. G.J. Houben, Faculty EEMCS[1], TU Delft
Dr. Martin Pinzger, Faculty EEMCS, TU Delft
Dr. Steven van Dijk, ORTEC Consulting Group, ORTEC

[1] Electrical Engineering, Mathematics and Computer Science

Preface

Since I was 8 years old I play the game of field hockey. For a few years I was also active as a coach and I developed an interest in the tactics of the game. Although I currently don't coach a team anymore the tactics of the game still interests me. During this thesis I was able to combine my passion for sport with my study.

During my research I went into the field and spoke to many inspiring people who are active on a high national and international level in field hockey. But one person stands out and that is Alyson Annan. Alyson introduced me to all the experts I have interviewed. Alyson, I really enjoyed our discussion about my research topic. Your knowledge and love for the game are really inspiring.

I want to thank my graduation committee for their supervision and guidance during my thesis work. First of all, professor Geert-Jan Houben for your guidance and feedback. Secondly Steven van Dijk, for the interesting pointers and our discussion about the architecture. And finally I would like to thank Martin Pinzger for being part of the committee.

This thesis is the final step of my study at the TU Delft. I would like to thank my family and friends for their support during my years at the TU Delft.

Arjan Peters
Delft, the Netherlands
November 28, 2012

Contents

List of Figures

List of Tables

1 Introduction

In sports a coach evaluates a match to increase the performance of the team. The insights gained from the evaluation can be used to provide feedback to the players, adapt the training practices and change the game strategy. The evaluation process is based on the observations of the coach. A shortcoming in match evaluation is the observational limitations of a coach [3]. It is impossible for a coach to view and remember every action in a match when watching the match because the view can be blocked and a match generally contains too many actions to remember them all. Furthermore, a coach can be distracted by his emotions and a coach typically focuses on things based on his prejudices. To aid a coach, video analysis is already widely adopted in field hockey. With video analysis, a match or moments of a match can be watched again. It allows a coach to evaluate a match better because by watching moments again he can observe things he missed during the match.

Commonly used video analysis software suites like SportsCode GameBreaker offer the ability to annotate a match. In match annotation, for events that happened during the match the occurrence time is tagged and labels can be added containing information about the events. Events that are currently tagged range in detail from penalty corners and shots on goals to every action with the ball. The label can contain information about the type of event, the player involved, and the spatial-temporal information when and where the event took place. A match annotation containing time and location data of the events is a historical spatial-temporal dataset. The match annotation simplifies the lookup of certain parts of a video of the match. The annotated events are not only relevant for linking to the video they also provide insight in the performance of a team. For example the frequency of a certain annotation like ‘goal attempt’ could be of interest for a coach. Match annotation data combined with data analysis and statistics offer new opportunities for coaches in field hockey in the match evaluation to increase the performance of the team.

The evaluation of match annotation data offers opportunities to get a better insight in the performance of a team. The challenge is not to collect the data, but the challenges are which data to collect and how to make use of the data by transforming it into applicable information for a coach [30]. The question what information is relevant is an important one: “not everything that counts can be counted, not everything that can be counted counts” [3]. Furthermore, coaches differ in their information needs. Coaches have different opinions on the importance of an event. An event can be judged by one coach as important while another coach can believe it is irrelevant. Besides the importance of an event the evaluation of an event also differs for each coach. The view of the coach is therefore important in the information need of the coach [33].

This thesis focuses on the development of a software system to support the analysis of match annotation data in field hockey to provide insight in the performance of a team. This chapter has the following structure; in the first section the background of data analysis in sport is discussed followed by the thesis objective in section 1.2. In 1.3 the structure of the thesis is discussed.

1.1 Background and motivation

In others sports there are a few success stories of using data analysis. An example is the attempt of Professor Anatoly Zelentsov with the soccer club Dynamo Kiev in the 1970s and 1980s. Using a quantitative approach and mathematical modeling he calculated the ideal strategy for every game which led to the winning of the UEFA Cup twice in that period [30]. Another example is the story "Moneyball" of the Oakland Athletics, one of the minor teams in the American Baseball League: by applying data analysis the team manager scouted cheap good players all the other teams missed resulting in reaching the playoffs year after year [23].

Performance measurement in sports can be split into several aspects of performance: technical, behavioral, physical and tactical [4]. Both tactical and physical performances are currently measured using specific tests, for example the speed and endurance of a player are physical performance metrics. Technical performance measures the ability of a player to perform a certain skill related to the game; an example is passing a ball, receiving a ball and dribbling with the ball. Currently available video analysis software offers the opportunity to evaluate the skill by watching a player perform that skill. The match annotation enables a coach to find the moments a player perform a certain skill more easily. The tactic of a match relates to the spatial-temporal aspect of the events [22]. What does a player do in a certain situation (location an time), and how does this relate to the overall tactic of the team. Tactic does not relate to a single event but to multiple events. The situation where an event takes place is determined by the previous events and the result of the event has effect on the future events; a temporal relation between the events exists. If a coach wants to get insight in the tactical performance by looking up a certain tactical pattern for evaluation it is related to the spatial-temporal aspects of the match annotation; a coach has to define a query on a spatial-temporal dataset.

1.2 Thesis objective

Current match analysis software packages have a limitation on evaluating tactical performance because they only enable a coach to look at single events and leave the spatial-temporal relations between the events uncovered. Tactical performance depends on the view of the coach. The goal of this work is to enable a coach to define and lookup tactical patterns to provide a better insight in the tactical performance of a team.

The *expressiveness* of the system is an important requirement. A coach should be able to declare a tactical pattern in the system. Therefore the system must be able to express a tactical pattern with concepts from the domain of field hockey.

Flexibility is another important requirement. A coach should be able to express and incorporate his view of field hockey in the system. A coach should be able to define every tactical situation that he wants to gain insight on.

In this thesis work a prototype will be designed, developed and evaluated to answer the following research question:

How to provide in a flexible manner insight in sport tactics to a coach using a query system for a historical spatial-temporal dataset?

To answer this research question the following objectives are formulated.

1. Requirement analyses for a software system to provide insight into sport tactics.
2. Design of a data model which enables the modeling of sport tactics.
3. Design of a query language to express a tactical pattern.
4. Design and implementation of a technical architecture of the prototype.
5. Design of a user interface that enables a coach to create a query and evaluate the results.

1.3 Thesis outline

This thesis has the following structure: in chapter 2 related works on querying spatial-temporal data and domain language design is covered, followed by the requirement analysis in chapter 3. Chapter 4 describes the data model which enables the modeling of sport tactics. Then, in chapter 5 the query language to describe tactics is discussed. The sixth chapter covers the technical architecture of the prototype and the user interface of the prototype is discussed in chapter 7. Chapter 8 is dedicated to the evaluation of the system. Chapter 9 is dedicated to the conclusion and gives directions for future research.

2 Related work

The goal of this thesis work is to enable a sports coach to query a spatial-temporal dataset in a flexible manner to provide a better insight in sport tactics. Therefore, research is reviewed that is related to this problem. Firstly, research in the field of querying a spatial-temporal dataset is reviewed. Secondly, research in the field of Domain Specific Languages (DSL) is reviewed. DSL's are a commonly used approach to enable end-user programming on a specific domain. In this work we want to enable a coach to define a tactical pattern and query it on a match annotation dataset. The final section contains a conclusion were a connection is made between the related work and the goal of this thesis.

2.1 Related research on querying a spatial-temporal dataset

In this section research on querying spatial-temporal datasets is reviewed. This is done in three parts. First, querying a temporal dataset is discussed, followed by querying a spatial dataset. Finally, existing spatial-temporal query systems are discussed.

2.1.1 Querying temporal data

In the querying of temporal datasets two different main approaches can be identified [1] :

1. Using first order logic
   In this approach time, is modeled in a timestamp column and relational calculus is used to query the temporal relations.

2. Using temporal logic
   Reasoning about temporal datasets can be done with temporal logic. Temporal logic is an area of formal logic to reason about propositions and predicates whose truth depends on time [20]. To enable temporal reasoning classic logic is extended with temporal connectives. Examples of temporal connectives are the operators Since and Until. These operators have the following definition:
   A *Until* B ➔ A must hold all the times until the time B holds
   A *Since* B ➔ A must have held all the time since B held
   [17].

   With Until-Since Logic [17] as with Linear-Time First Order Temporal Logic [20] the following operators can be derived from the Since and Until operator: *Past, Previous, Always in the Past, Future, Next, Always in the Future.*

Both approaches enable the querying of a temporal dataset and have the same expressiveness [1]. The temporal logic based approach has the following 2 advantages over the one based on first order logic:

1. Temporal logic makes the notation of time more implicit; therefore, it reduces complexity because of its reasoning with temporal operators instead of using the timestamp column [1].
2. Temporal logic offers a richer notation [7] which make the temporal relations between entities more explicit; therefore, it eases the reasoning with temporal entities.

An advantage of first order logic over temporal logic is that there are numerous approaches for temporal databases and query languages, but none of them is a well accepted solution that made it to the market [36]. Using first order logic, conventional databases and query languages can be used.

2.1.2 Querying spatial data

To query spatial data we want to reason with spatial entities. With spatial relations reasoning can be applied with two spatial entities. Both Pelekis et all [27] and S-TVQE [5] classify three types of relations:

1. Directional (Above, RightOf, LeftOf etc)
2. Metric (Length, Distance, Angle etc)
3. Topological (Meet, Inside, Overlap etc)

Directional relations describe the orientation and directional features between two objects in a space. A directional relation is visualized in Figure 1. A metric relation gets a value of a spatial object or between two objects like the length or perimeter. Topological relations can be useful to derive if an event takes place at a certain location. Using spatial relations more information from an object can be derived.

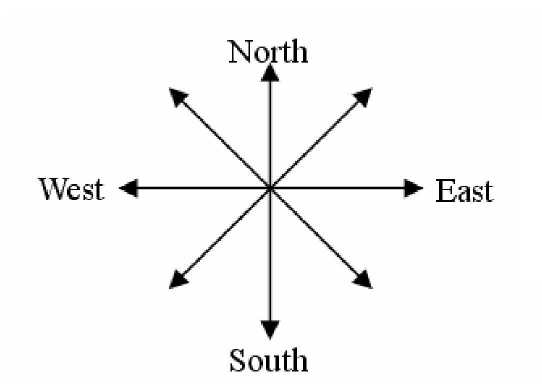

Figure 1: Spatial relations [24]

2.1.3 Spatial-temporal Query Systems

It is considered desirable to express a query in a spatial-temporal database in a higher level of abstraction than SQL using concepts from natural languages [36]. To achieve this, there have been multiple textual approaches. For temporal datasets new query languages are developed like TQuel and TSQL2 [7]. For spatial reasoning, extensions on SQL are developed such as Spatial SQL [9]. ∑QL is a query language that supports both spatial as temporal constructions [6]. These languages are extensions of SQL and require experience with SQL and the relational model. In spatial-temporal datasets visual query interfaces are more often applied. This has the following reasons:

1. Spatial data are inherently visual [9]
2. Query languages for spatial-temporal datasets are more complicated because they have to be able to express the temporal and spatial relationships [5]

An example of a visual query interfaces for spatial-temporal datasets is S-TVQE. S-TVQE works on a conventional database using standards as SQL and OpenGIS. Using a visual interface a user is able to specify both the temporal and spatial dimensions [5]. On top of ∑QL there is also a visual interface developed aiming to enable a user with no knowledge of query languages to query the spatial-temporal dataset [24]. An example of a temporal visual query interface is the Temporal Database System [21], instead of creating SQL queries the querying is executed in memory.

To enable a coach to perform tactical analysis using a spatial-temporal dataset was also done by Kuijpers in his thesis work. Instead of field hockey the domain was tennis. He designed a specialized query language to create SQL queries using first order logic on a timestamp column to enable temporal reasoning because of the complexity of the syntax of temporal logic. However, the problem with his system was the complexity of the designed language. Even without temporal construction his language was too complex for coaches and therefore he suggested a visual interface as future work to improve the usability of creating a query. [22]

The described systems all use a form of a temporal mediator [36]. The process of querying happens in the following 3 layers:

1. Visual interface
2. Temporal mediator
3. Database

In the visual interface the query is entered by the user. The temporal mediator acts as a computational layer between the user interface and the database and performs the temporal reasoning. The database is responsible for the persistence storage of the data.

A challenge in visual query interfaces is that the system aims to be both versatile and user friendly but those two are often at odds with each other. Functionalities that offer lot of expressive power and flexibility have a negative impact on the intuitiveness of the user interface [11].

2.2 Domain Specific Language

To enable users from a particular domain to interact with the spatial-temporal dataset a solution can be found in the field of DSL. "A DSL is a programming language that offers through appropriate notations and abstractions expressive power focused on and usually restricted to a specific domain" [8]. A DSL is a programming language; knowledge by humans is expressed in such a way that it can be executed by a computer but a DSL is still easy to understand by humans (lay language). A DSL has limited expressiveness; it only supports what is relevant to the domain, while a general purpose language like C++ offers way more functionalities because a general purpose language is Turing complete. A DSL has a domain focus, using a high level of abstraction it hides the complexity of the irrelevant parts of the system and only focuses on the domain. A DSL has the following advantages and disadvantages [16]:

Advantages

1. Improve development productivity
   Using a DSL complex programming structures are hidden using a higher level abstraction which increases the productivity of a programmer, and reduces the amount of errors in the code.
2. Communication with Domain Experts
   Both the programmer and the domain expert communicate in the same language.

Disadvantages

1. New language to learn
   When implementing a DSL the users need to learn the new language first before they can work with it.

This disadvantage is not relevant in this thesis because coaches have little to none experience with programming languages and have to learn the language either way. A DSL has the advantage that it is easier to learn than a general purpose language because of the domain focus and the limited expressiveness. [16]

DSLs can be implemented in two different ways, as an external or as an internal DSL. With an internal DSL, the DSL is created within the syntax of an existing general purpose language. An external DSL is a separate language using its own syntax. An external DSL requires more effort to implement because a parser needs to be created but it has the advantages that the syntax is not restricted to a general purpose language which holds for an internal DSL. An example of an internal DSL is the jQuery library which supports the interaction with DOM elements and AJAX in JavaScript. Examples of external DSLs are XAML and regular expressions. An example of a temporal DSL applied in the sport domain is EasyTime [15]. EasyTime is a DSL used for controlling time measurement sensors and writing events to a database. Another example is the thesis of Kuijpers where an external DSL is created to analyze the tactics in tennis [22].

2.3 Conclusion

A goal of this thesis is to enable a coach to query for a tactical pattern on a spatial-temporal dataset. Therefore in this chapter spatial-temporal query languages and systems are discussed. A tactical pattern is a sequence of actions linked together by a temporal relation. A temporal relation can be expressed using temporal logic or first order logic on a timestamp column. Temporal logic makes time more implicit and exposes the relation between temporal entities in a more explicit way which results in a richer expression and therefore eases the reasoning with temporal entities and has therefore an advantage.

Next to the temporal component the spatial component is also important in evaluating a tactical pattern for a coach. A team can have different tactics depending on the location in the field. Besides the location of an action the distance can be derived between two actions which can be used to describe the length of an action like a *long pass*. A characteristic of a tactical pattern can be that it only consists of long passes. Spatial relations offer the possibility to derive Directional, Metric and Topological information.

To create a spatial-temporal query there a numerous query languages developed which support spatial-temporal reasoning. A problem with these approaches is that they are extensions of SQL and therefore complex because a user needs to understand both the relational model as the spatial-temporal extension. A solution for this problem can be found in the construction of a DSL. With limited focused expressiveness on the field hockey domain it can enable a coach to better interact with system. To express a tactical pattern the DSL must be able to express spatial-temporal relations.

3 Requirements

In this chapter, the requirements of a system that enables a coach to query for tactical situations on a match annotation file are described. The requirements are leading into the development of the prototype. The process of the requirement analysis is visualized in Figure 2. The first step in the process is the domain study; the domain study is performed to gain more insight on the type and structure of the tactical questions coaches are having. The result of the domain study is a number of representative example tactical questions. Those example questions are then analyzed to create the functional requirements that enable the system to answer the example questions. The domain study also provides insight in the current process of match analysis of coaches. This insight is also used to define the requirements.

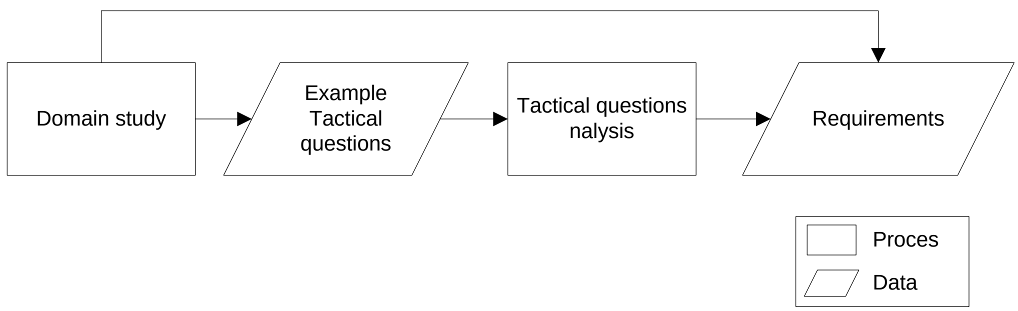

Figure 2: Requirement analyses process

This chapter has the following structure: in the first section the domain study and the example tactical questions are discussed. In the second section the example questions are analyzed. The final section describes the requirements.

3.1 Domain study

This section is dedicated to the analysis of tactical questions that coaches are having. Discussions and brainstorm sessions have been held with multiple coaches to gain better insight in the tactical questions. The sessions were of an exploratory nature because there are large differences in the experience that coaches have with data analysis. The starting point was always the video of a certain match and the match annotation because that is the current process of match analysis that coaches apply. Analyzing multiple matches at once is at this moment not possible with the existing tools. Interesting tactical situations then were defined that with the current match annotation and video analysis software could not be found. In the domain study five coaches participated which are active on a high national and international level.

The coaches had different types of questions which were often related to specific players or situations in specific matches. Former field hockey player and two times Olympic gold medalist Alyson Annan is an expert and pioneer on data analysis in field hockey in the Netherlands and she was present and supported every session with the other coaches. Together with Alyson Annan the following tactical question were selected that are considered representative:

1. How does a team penetrate the circle?
2. How is a goal attempt created?
3. Is it true that the majority of the passes into player X are diagonal upfield?
4. Where does a team X gain ball possession?
5. In Figure 3 the outlet of the British women team is sketched. The first action (white dot 1) is a pass to the defending midfielder (Kate Walsh). The second action (yellow dot 1) is a movement the ball which is followed by a pass upfield (yellow dot 2). Is this happening and how often during a match?

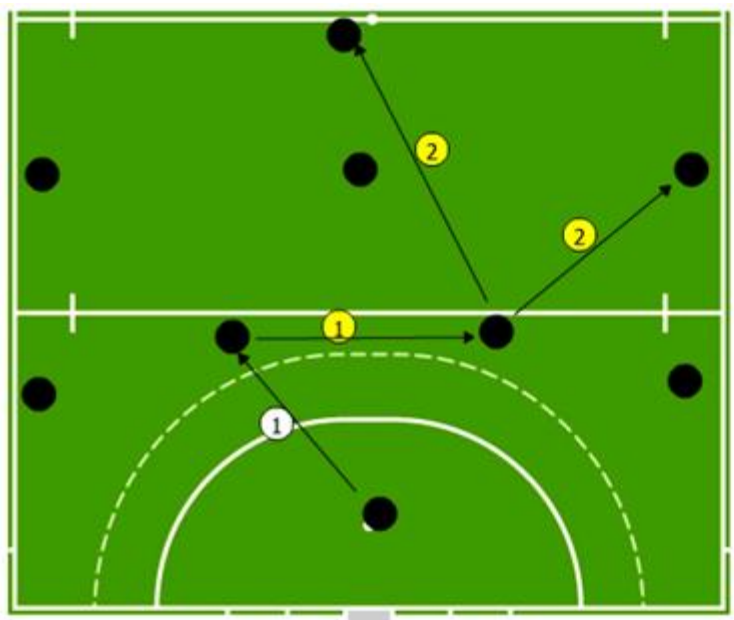

Figure 3: Example tactical outlet

Most questions are clear but some questions are up for debate. For example the definition of ball possession differs between different coaches. Some coaches see ball possession as a handling of the ball while others see it has a handling which is either preceded or followed by an action of a team member. Furthermore, the definition of the spatial region like the left-field differs for some coaches. Spatial regions defined by lines on the field like the circle is no point of debate but left-field is according to some coaches, the region in the field 5 meters from the left sideline while other use a definition of 10 meters for example.

3.2 Analysis tactical questions

The next step is the analysis of the example question created in the previous section and the extraction of the functional requirements from those questions. By analyzing the structure of the questions properties from the questions are extracted. The extracted properties are the components which joined together form a tactical question. From the question the following question components are extracted and explained in Table 1.

Table 1: Question components

Property	Question	Description
1. Subject	1, 3, 4, 5	Player, Team or position in a team line up performing the action.
2. Type of action	1, 2, 5	Type of ball handling of the action like pass, goal attempt
3. Spatial Direction	3, 5	Direction of play like diagonal, upfield, length of the action
4. Spatial Location	1, 2, 4	Location where an action is performed or where an action transfers the play to
5. Temporal origin	1, 2	What happened before the action (how do we get to the situation?)
6. Temporal result	5	What happens after an action (what is the result of the action?)

Another issue with the questions is the type of answer coaches are expecting from different questions. Looking at example question 1 and 2, a coach wants to have insight in the actions preceding a certain action (goal attempt / circle penetration); the preceding actions are asked. While in example question 3 and 5, coaches are interested whether a certain situation occurs, and if so they want to see it. Finally, the third question asks for a spatial location. During the interviews coaches often used a video fragment to explain a situation. It showed how the identification (selection) and viewing of video fragments is part of the analysis.

3.3 Functional Requirements

The operational process of a tactical match analysis system can be split into 4 different phases. First the scope is set; the matches that are going to be analyzed. The second step is creating and executing of a tactical query. The third step is visualization of the result. Afterwards a coach can modify his query with knowledge gained from his interpretation of the query result. The fourth process is related to the flexibility requirement of the system; a coach should be able to incorporate his own tactical view in the system.

1. Define scope

The match analysis currently starts with two things: a video file and a match annotation. Coaches analyze match by match and cannot analyze multiple matches at once because of limitations of the current software. Although some coaches already see the advantages of analyzing multiple matches at once most focused on a single match.

The scope definitions has the following main functional requirements

   a. Open one or more match annotations files
   b. Align a video with the match annotation file

2. Creating and executing query

The creation of the query is done by using the question components described in the previous section. With these components a tactical question can be formulated. This query can then be executed by the system. Some questions are not match specific but can be applied to multiple matches. Therefore, it is desirable to save questions so they can be used again. The creation and execution of a tactical query has the following requirements

   a. Create a tactical query by adding question components (subject, type, spatial, temporal)
   b. Executing a tactical query
   c. Save, update, open, delete a tactical query

3. Visualization of the result

The thrid step is the visualization of the result of the query. Different types of results are desirable depending on the type of the questions. The requirements of the visualization of the results are:

   a. Showing the related video fragment
   b. Visualizing the spatial data
   c. Visualizing the temporal data
   d. Showing the corresponding action in a tabular visualization

4. Create + edit definitions

Flexibility is an important requirement of this thesis; a coach must be able to incorporate his own definitions in the system. An example of such a definition is a spatial location like the left-zone, because they can differ for every coach. Other examples are the length and directions. The requirements related to the definitions are

   a. Save, update, open, delete a coach definition

Delineation

One of the goals of this thesis is the explore how to enable a coach to query a match annotation file to evaluate tactical situations, not to create a full functional match analysis software package. In the development of the prototype only the parts that are necessary to evaluate the querying of tactical situations by a coach will be developed. The requirements that are out of scope will not be implemented in the prototype but in the overall design those are taken into account. The requirements that are out of scope are:

1. Definition of the scope
   The prototype will work with one predefined match and aligned video file
2. Save, update, open, delete a tactical query
   These functionalities ease the use for recurrent use, which will not be the case with the evaluation of the prototype.

4 Domain model

In the previous chapter we defined properties of actions that can be queried. Based on these requirements a domain model is created. The goal of the domain model is to model the domain-specific elements and express a tactical pattern. This chapter has the following structure: in the first section the temporal relations are discussed followed by the spatial relations. The third section describes the domain model.

4.1 Temporal relations

When evaluating the tactics of a game a coach is not interested in a specific action but in a combination of actions like the buildup towards scoring a goal, or how specific actions follow upon each other. In current match annotations, the actions are modeled as separate event entities which all have a timestamp. When a coach wants to know what the following action is he has to derive the following action by looking for the next action based on the timestamp. The information is available in the data but only implicit. In the domain model the data is modeled at a higher abstraction level and this temporal relation can be made explicit. Instead of looking at a single action an action has a predecessor and a successor. The predecessor is the action that happened before the action and the successor is the action that happens after the action. This is defined by the timestamp of the actions. Exceptions are the first action and the last action of a period. A match consists of two periods of 35 minutes. The first action of a period does not have a predecessor while the last action of a period does not have a successor. This enables reasoning about the temporal relations between the actions while hiding the timestamps. The temporal relation between the actions is visualized in Figure 4.

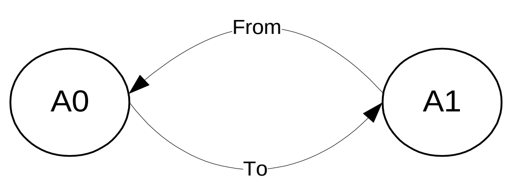

Figure 4: Temporal relation between two actions

4.2 Spatial relations

An action occurs at a certain location in the hockey field. The same as with the temporal relations the relations are made explicit in the domain model. This is done by deriving the spatial relations from the match annotation and mapping it to a class that describes the relations. The classification is performed based on the definitions of a coach and can be configured by a coach. This supports the flexibility requirements of the system. The process of the classification of the spatial relations is visualized in Figure 5.

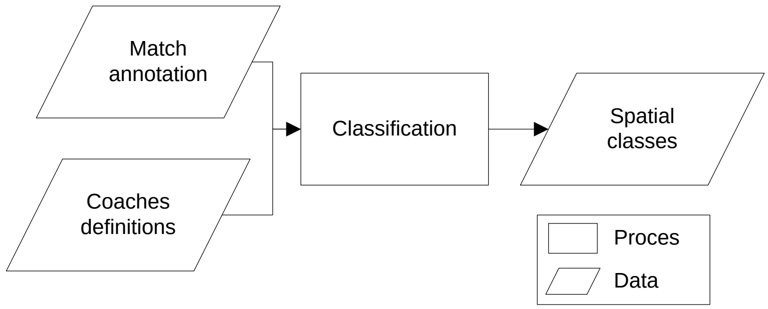

Figure 5: Spatial relations classification process

Three different types of spatial relations are derived based on the related work covered in chapter 2:

1. Directional
2. Metric
3. Topological

A topological relation describes the location where an action takes place. In the match annotation the location is described using a coordinate system. Using the topological relation this can be mapped to a region or area of the field; the action takes place inside the circle for example. By hiding the spatial location annotation and using classes to describe areas the complexity for a coach is reduced because he does not have to learn the location annotation. A location can be mapped to multiple areas like left-zone and attacking-zone.

Metric and directional relations between the locations of two actions are depended on the temporal relations between two actions. Those relations describe a movement of the ball or the player. A distinction can be made between the types of action within field hockey; some actions are happening in motion, or describe a movement of the ball while other actions are stationary. In the action set a distinction can be made between static and dynamic actions. For these different types of action the following definitions are used:

Static Action: *is an action when there is no movement by the player/ball, a receiving action for example.*

Dynamic Action: *is an action when there is movement by the player/ball, a pass for example.*

In this thesis work the match annotation methodology "Effectivity in Action" (EIA) [32] is used as an example case (see appendix A). In EIA the Action Types are divided between static and dynamic actions.

Static Action Type:

- *Interception*
- *Defending action*
- *Receiving*
- *Goal Attempt*
- *Penalty shot*
- *Save on goal attempt*
- *Save on penalty corner*
- *Save on penalty shot*
- *Penalty corner stop*

Dynamic Action Type:

- *Pass*
- *Attacking action*
- *Dribble*
- *Free hit*
- *Free hit self pass*
- *Penalty corner pass*

For a Dynamic Action both metric and directional relations can be derived. A metric relation like the distance between two locations is a number. Using a classification a range of numbers can be mapped to a class, for example 0 to 5 meters is a *Short Distance*. Using this classification a coach does not have to reason in numbers. For directional relations the same idea is applied. Again using a classification the angles are mapped to classes. Examples of these classifications are *Deep, Diagonal and Wide*.

4.3 Domain Model

In chapter 3 requirements were set for the properties of an action to describe a tactical question. These properties are:

- Subject
    - Player, Position, Team
- Spatial
    - Distance, Area, Direction, Angle
- Temporal
    - Origin, Result
- Action Type

Using the described spatial-temporal relations in the previous two sections and combining this with the requirements of chapter 3 the domain model is formed. In Figure 6 a class model visualizes the domain model. On the left side in the yellow square the subject properties are modeled. An action is performed by a player and a player has a position like defender or midfielder and a player is part of a team. The right blue square shows the derived spatial relations. The spatial derivates distance, direction and angle can only by derived from dynamic actions. The actions are linked together using a From and To relation, which makes the temporal relation between the actions explicit.

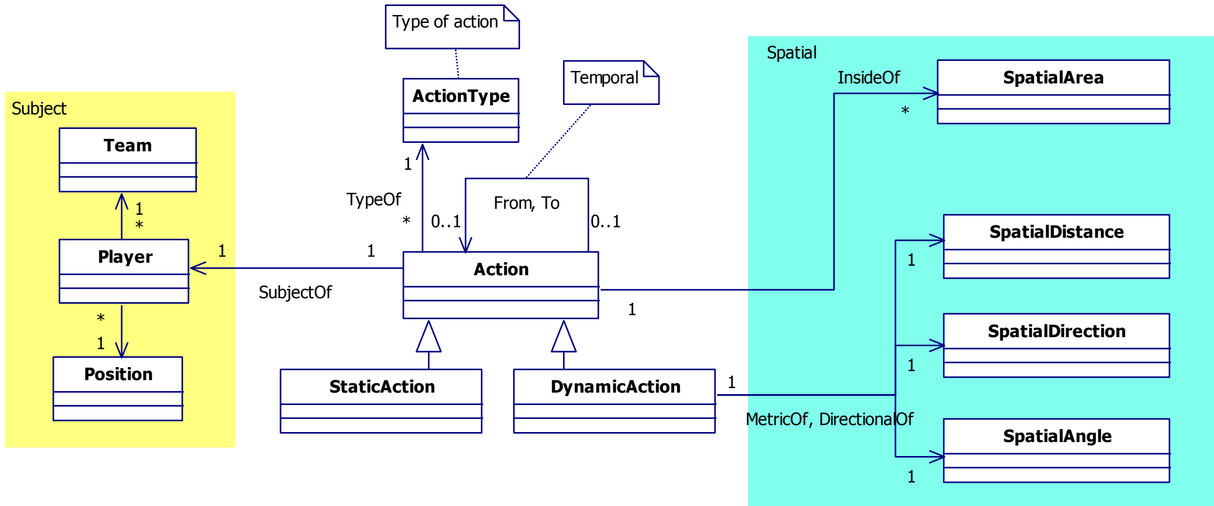

Figure 6: Domain model

A task of the domain model is to express a tactical pattern. Via the domain study the components necessary to describe a tactical pattern are derived. These components are represented in the domain model. Therefore the domain model contributes to the expressiveness criteria of the system because with this domain model a tactical question can be expressed.

In the domain model a level of abstraction is created. By making the spatial and temporal relations explicit as properties of an action the system offers a richer expression of an action. This enables the expression of domain concepts of field hockey like a "*pass by player A to player B*". A downside to this abstraction is that the system has a lower expressive power because not every possible question involving the spatial and temporal information can be formulated anymore.

Looking at flexibility criteria, the domain model offers the possibility for a coach to configure the spatial classification. This configurability enables a coach to create his own definitions for example the spatial lengths.

5 Domain language

In the previous chapter the match annotation data is enriched to express spatial and temporal relations explicit and the data is modeled at an higher abstraction by hiding the spatial and temporal notations. This domain model offers the ability to express a tactical pattern at a higher abstraction but it still requires programming knowledge to operate. In this chapter a DSL is proposed which enables a coach to express a tactical pattern at the conceptual level. The DSL acts as an instrument to transfer the knowledge of a coach to the system. The DSL can be split into two parts: the semantics and the syntax. The semantic describes the meaning of the DSL and the syntax the form of the DSL. This chapter has the following structure: in the first section the semantics are covered followed by the syntax in the second section. In the third section is dedicated to a conclusion of this chapter.

5.1 Semantics

The section covers the semantics of the DSL. The goal of the DSL is to express a tactical pattern. With the semantics the constructs are described which enable the expression of a tactical pattern on top of the domain model. A tactical pattern is a sequence of actions. To express a tactical pattern two steps can be indentified:

1. Express a single action
2. Combining actions to express a tactical pattern

5.1.1 Express a single action

The first step is creating a specification of an action. In the specification the conditions which an action has to satisfy are described. This is done by expressing the valid attributes of an action. A coach must be able to express a combination of attributes. For example a coach wants to see the passes of two players. To express such a question Boolean constructions are necessary. The requirements for the expression of a single action are therefore: defining valid attributes of an action and combining valid attributes using Boolean logic.

Creating specification that describes an object is a common case that can be found in business systems for selection and validation scenarios [12]. Using business rules, a range of values is defined that determine whether an object meets the specification or not. Evans and Fowler describe the specification pattern [13]. This design pattern creates a separation between the domain object and the object containing the business rules describing the object, see Figure 7.

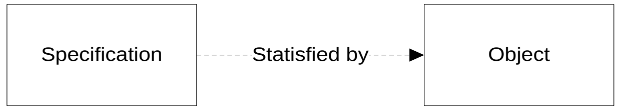

Figure 7: Object model specification pattern [12]

A specification is a predicate which determines whether an object satisfies a criterion or not [12]. Figure 8 visualizes the specification pattern. The specification pattern support the Boolean operators AND, OR, and NOT. With the specification pattern a specification can be chained together with another specification using Boolean logic.

To specify an action a specification for that action has to be created. This is done by creating an ActionSpecification. The specification of an action is a collection of specifications describing the attributes (Player, ActionType etc) of an action. To describe an attribute of an action like the Player performing the action a specification describing the player has to be added to the ActionSpecification.

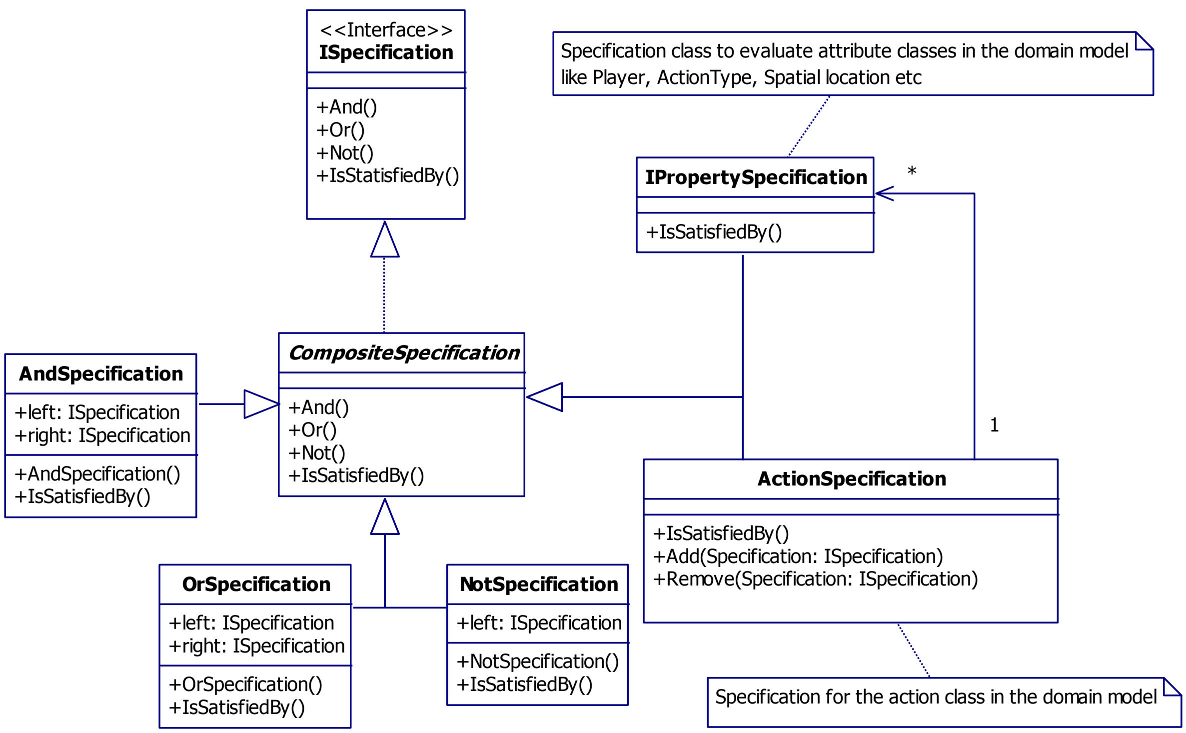

Figure 8: Class diagram specification pattern

5.1.2 Express a tactical pattern

A tactical pattern is a list of actions linked together by a temporal relation. A tactical pattern can be used to look for a cause (reason back in time) or for a result (reason ahead in time). Two different types of tactical patterns can be identified:

1. Sequence
   A list of actions linked to the preceding or consecutive action
2. Interval
   A list of actions defined by a start and an end action.

To create a sequence or an interval of actions temporal logic is implemented. To enable the expressing of a tactical pattern the specification pattern is extended with temporal specifications. The following temporal specifications are implemented: Since, Until, FollowedBy, PrecededBy. These specifications work between actions because a temporal relation exists between the actions. Table 2 gives an overview of the temporal specifications.

Table 2: Temporal specifications

Specification	Type	Future / Past
FollowedBy	Sequence	Future
PrecededBy	Sequence	Past
Since	Interval	Past
Until	Interval	Future

The Since and Until specifications define an interval of actions. An interval is visualized in Figure 9. An interval specification consists of three ActionSpecification:

1. Start specification
   Specification defining the start of an interval
2. End specification
   Specification defining the start of an interval
3. Not specification
   Specification which not holds during the interval

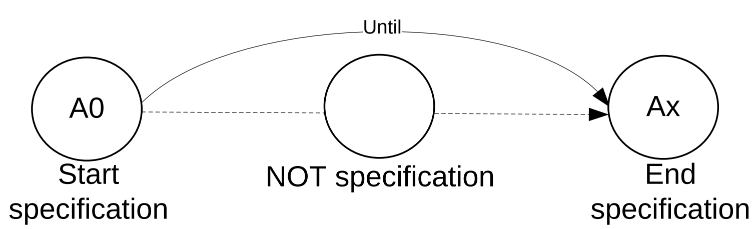

Figure 9: Interval specification

5.2 Syntax

With the specification pattern from the previous section a specification describing a tactical pattern can be expressed. But this still requires knowledge of a programming language because all the specification objects need to be initialized. The focus of the syntax of the DSL is the communication with the coach. Therefore the syntax must be understandable for a coach but still have the appropriate expression power to express a tactical pattern.

To enable a coach to work with the DSL the language is inspired by natural language constructions. When describing an action coaches tend to use adjectives to describe the properties of an action. Coaches talk about a *Long Pass* for example. A *Long Pass* has to be converted to the specification of an action; an action with the type pass and the length long.

To simulate the use of adjectives when creating a specification in the DSL method chaining is applied via an API which creates the specification. With method chaining, multiple method calls are chained together into a single statement. The expressiveness doesn't come from a single method call but also from the combined method calls. Method chaining provides a language-like fluent way of defining an statement and therefore offers readable expression [16].

Similar as in the previous section the distinction between expressing a single action and combining action into a tactical pattern is made in the description of the syntax.

5.2.1 Express a single action

Table 3 gives an overview of the methods to specify the attribute of a single action. An example of a specification of an action is:

```
all().ByTeam(NederlandDames).Type(Pass).Length(Long)
```

The beginning statement is colored blue and initializes the specification of an action. The methods for adding a specification of an attribute of an action a colored green and the specification of an attribute are colored red.

The specification parameters in Table 3 describe a corresponding specification for that attribute. This specification can contain a Boolean expression. For example a coach wants to see the passes of two players. This results in the specification: PlayerA OR PlayerB. But research shows that Boolean expressions are difficult for non programmers, both slow to formulate and error-prone [19]. AND and OR constructions in Boolean logic have a different meaning in English: AND is exclusive in Boolean logic but tends to be inclusive in English and OR is inclusive in Boolean logic but tends to be exclusive in English.

Table 3: Methods to specify an action

Method	Specification Parameter	Semantic
All		Initialization of an ActionSpecification
InLocation	Spatial Location	Location where an action takes place
ByPlayer	Player	Player performing the action
ByTeam	Team	Team performing the action
Type	ActionType	The type of action
Direction	Spatial Direction	Spatial direction of the action
Angle	Spatial Angle	Spatial angle of the action
Length	Spatial Length	Spatial length of the action
ToLocation	Spatial Location	Location of the following action
ToPlayer	Player	Player performing the following action
ToTeam	Team	Team performing the following action
FromLocation	Spatial Location	Location of the preceding action
FromPlayer	Player	Player preceding the following action
FromTeam	Team	Team preceding the following action

To enable a coach to work with the language different constructions are created for the expression of Boolean logic in the specification of the attributes of an action. Table 4 gives an overview of the syntax of Boolean logic in the expression of the attribute of an action.

Table 4: Syntax of Boolean logic in the expression of an attribute of an action

Boolean operator	DSL	Example
AND	Dash -	A-B
OR	Comma ,	A, B

An example of a specification using Boolean constructions is:

```
all().ByPlayer(Welten, Verhagen).Type(Pass). InLocation(Left-Defending23)
```

In this example a OR construction can be found in the specification of the players and a AND construction can be found in specification of a location.

5.2.2 Express a tactical pattern

To create a tactical pattern actions have to be chained together using the temporal operators: Since, Until, FollowedBy, PrecededBy. The Since and Until method require two parameters. One for the ending action specification and one for the specification that is not valid during the whole interval. This results in the following example expression:

```
all().ByTeam(NederlandDames).Type(Pass).InLocation(Defending23)
.Until(
      all().InLocation(Attacking23), all().Type(Foul)
      )
```

This query is the specification of the outlet of a team; the transition from a pass in the defending area until the ball ends up in the circle without a foul. For advanced queries between action specifications Boolean specifications are also enabled in the internal DSL.

5.3 Conclusion

The goal of this chapter was to design a DSL which enables a coach to express a tactical pattern with concepts from the domain. To express a tactical pattern a coach must first be able to express and action by specifying the valid attributes of that action. Afterwards actions can be chained together using temporal operators. Using the specification pattern a specification of an action can be created. This solution offers a rich expressiveness because specifications can be chained together using Boolean operators. To support the temporal reasoning the specification pattern is extended with temporal operators.

The specification pattern offers the expressiveness but it is not on the conceptual level of a coach. The DSL offers the expression of a tactical pattern via method chaining. Method chaining offers a language like fluent way of defining an action. Boolean operators are replaced with construction from written language because they can be considered complex and confusing for coaches. The combination of method chaining and replacing Boolean operators eases the communication with the coach.

6 Technical Architecture

This chapter presents an architecture for a system to query for a tactical pattern. This architecture consists of three distinctive layers: Data Layer, Query Layer, and Visualization Layer (see Figure 10). Each layer represents a specific set of functionalities. The data model and the DSL from the previous chapters are integrated in this architecture. These layers combined form the system. This chapter discusses the functionalities and implementation details of the Data Layer in the first section and the Query Layer in the second section. The Visualization Layer is discussed in the next chapter.

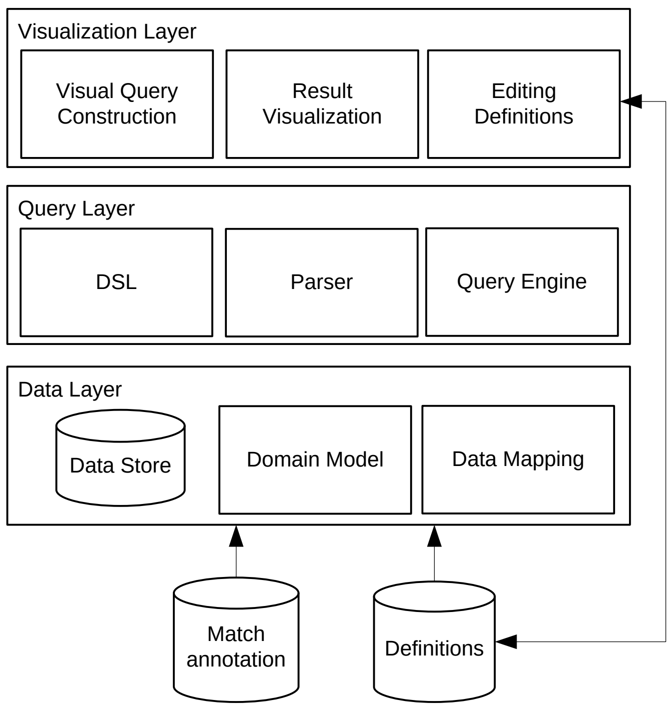

Figure 10: Architecture

6.1 Data Layer

The Data Layer holds the Domain Model. The match annotation data is loaded and the Domain Model is created by the Data Mapping process. The populated Domain Model is stored in an InMemory Data Store which can be accessed by the Query Layer. The Data Layer hides the match annotation data and the Data Mapping process for the rest of the system.

Data Mapping

A global overview of the data mapping process can be found in Figure 11. The definitions can be split into sport specific definitions and coaches' definitions. The sport specific definitions contain information about the match annotation specific for a sport. An example is the type of actions that are available in a sport and how the actions based on the action type are split into static and dynamic actions which is used to determine if spatial relations can be derived. The coaches' definitions contain information specific for the coach.

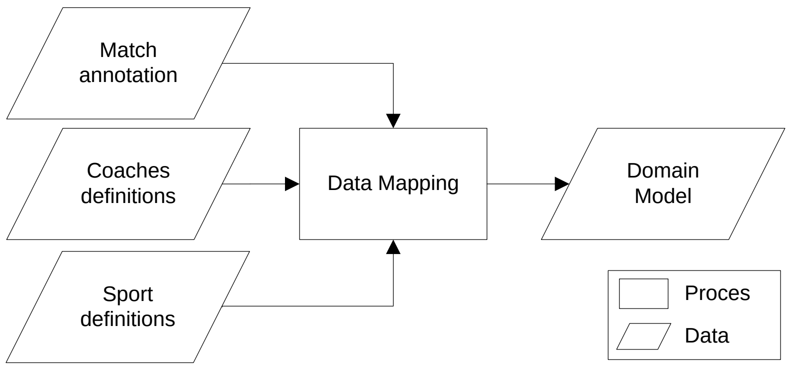

Figure 11: Data Mapping process

The coaches' definitions describe a spatial relation in the match annotation and the corresponding representation in the Domain Model. The spatial location of an action in the match annotation is modeled as an XY coordinate (see Appendix A). The XY coordinate can be mapped to an area on the field based on a definition. The process checks if an action occurs within the area defined by the definition. An example of an area is Left-zone. The location definition consists of one or more shapes which defines an area. These shapes are rectangles and circles. To define the circle in field hockey a combination of shapes is required. A circle in field hockey consists of two circles and a rectangle [35]. Figure 12 visualizes the spatial regions defining a field hockey circle. The area outside the circle visualized in Figure 12 is called the 23meter area in field hockey. To describe the 23meter area the area's describing the circle area has to be excluded. The mapping for a location has the following definition:

Location: *Spatial area defined by one or more shapes (Rectangles/Circles) which define an area to be included and a list of shapes which define an area to be excluded*

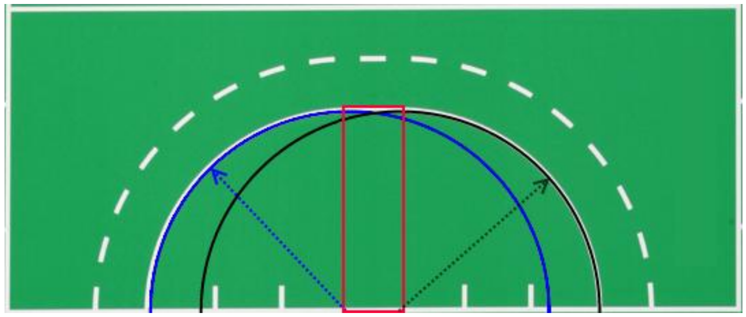

Figure 12: Field hockey 23meter zone and circle

For a dynamic action the length, angle and direction of play can derived. This is achieved by looking at the consecutive action. A problem with the spatial data is the orientation of the match annotation. The spatial data is relative to the team performing the action based on the direction of play looking from the perspective of the goal keeper (see Appendix A). When classifying a length or angle of an action a problem arises when the consecutive action is from the opponent team because the spatial data of the consecutive action is from a different orientation. Therefore the data mapping process needs to mirror the spatial data when the consecutive action is from the opponent team. For the length and angle the mapping has the following definition:

Length: *Euclidian distance between two points described by a minimum and a maximum value*

Angle: *The geometric angle described in grades with a minimum and maximum value*

Direction of Play: *A range of geometric angles described by a minimum and maximum value*

The definitions are stored in an XML file.

```
<Lengths>
   <LengthMapping>
     <Pointer>
       <Name>Short</Name>
     </Pointer>
     <Min>0</Min>
     <Max>10</Max>
   </LengthMapping>
</Lengths>
```

This XML fragment describes the spatial length *Short*. A length has a lower bound (min) and an upper bound (max) which is used to determine if a distance between two actions matches this classification. The use of the definitions enables the system to be flexible for both the view of the coach and for multiple sports.

6.2 Query Layer

The query layer is responsible for the information retrieval. The layer interacts with the Data Store of the Data Layer. A part of this layer is the DSL described in chapter 5 where a query is in constructed. In the DSL a user can express a tactical pattern. To retrieve the information from of the Data Store the query needs to be parsed an executed.

Query Parsing

A query is a string in the DSL describing a tactical pattern. The system requires that a user can create a query during runtime and execute it. Therefore a parser is needed to parse the query string. The parser is responsible for transforming the string into a specification-object which can be evaluated.

Using a script engine the query can be parsed and executed during runtime. For the script engine IronPython is used. IronPython is a .NET implementation of the general purpose language Python. With IronPython functionalities from both Python and .NET libraries can be used. IronPython is a dynamic language and can execute uncompiled code (scripts) during runtime. IronPython is offering a great flexibility because code can be created and executed during runtime. [25]

In the script engine the .NET libraries of the DSL (chapter 5) and Domain Model (chapter 4) are loaded. From the Data Store the unique instances are retrieved from the different attributes of an action like the lengths. For these instances specifications-objects are created and loaded in the scope of the script engine (see Figure 13). This enables that a specification can accessed using its variable name, like *Long* for the specification of spatial length with the name long as in the following query:

```
all().Length(Long)
```

Query Engine

The Query Engine is responsible of the evaluation of the data in the Data Store based on the specification provided by the Parser. Using a sequence diagram the interaction between the different components is explained. The sequence diagram can be found in Figure 14.

The first step is the parsing of the query by the script engine. The parsing step converts a string written in the DSL of chapter 4 in an ActionSpecification (step 2). The ActionSpecification is passed through to the Query Engine (step 4). The Query Engine evaluates the data in the Data Store. A QueryResult object is created which holds the action or interval of actions that meet the specifications. The QueryResult is returned to the user.

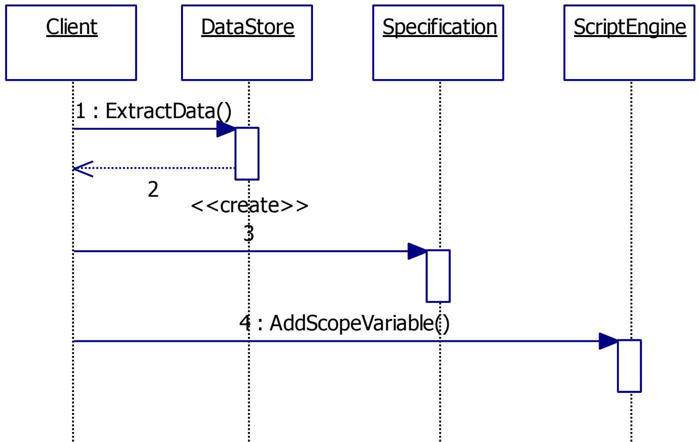

Figure 13: Sequence diagram scope initialization

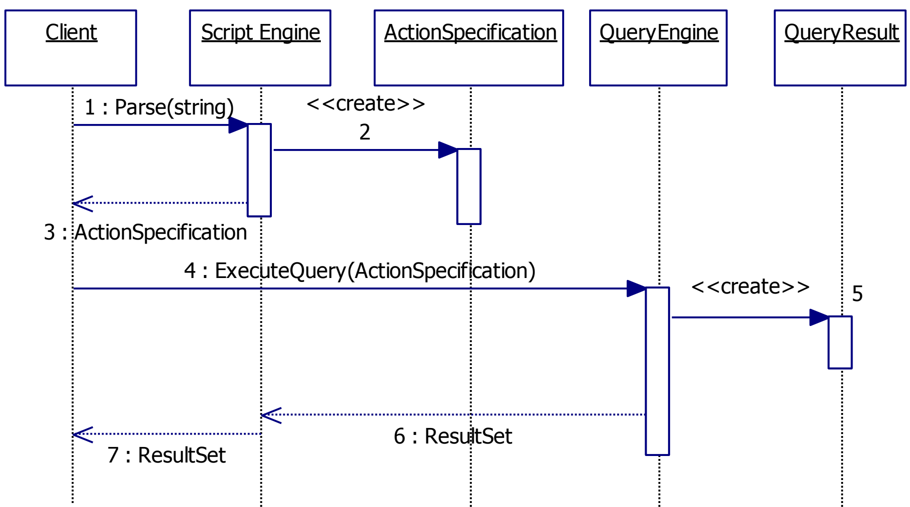

Figure 14: Sequence diagram Query Execution

7 User interface

The goal of the user interface is to enable a coach to create a tactical query and interpreted the result from a query to gain insight in a tactical pattern. The user interface is constructed on top of the query layer discussed in the previous chapter. An important aspect of the user interface is that it provides an easy to use access to the query system. By using a visual query interface and hiding the DSL syntax it eases the communication with the coach. By visualizing the result the system provides insight in a specific tactical pattern.

This chapter has the following structure: in the first section a global overview of the user interface is discussed. Section 7.2 is dedicated to the visual query construction. The third section discusses the visual representation of the results of a query. In the final section the visual editor of coaches' definitions file is shortly discussed.

7.1 Overview

A global overview of the user interface can be found in Figure 15. The interface contains multiple regions with different functionalities. The regions are numbered in the image and have the following short explanation:

1. Result overview
   The possibility to select a result for further inspection by other result visualizations and gives the number of results of a query.

2. Spatial visualization
   Visualizes the spatial component of a selected result

3. Tabular visualization
   Shows the actions related to a result for a selected query result

4. Query construction
   The query construction component contains both a visual query builder as a textual query form.

5. Video preview
   Video player to play the corresponding video fragment related to a selected result.

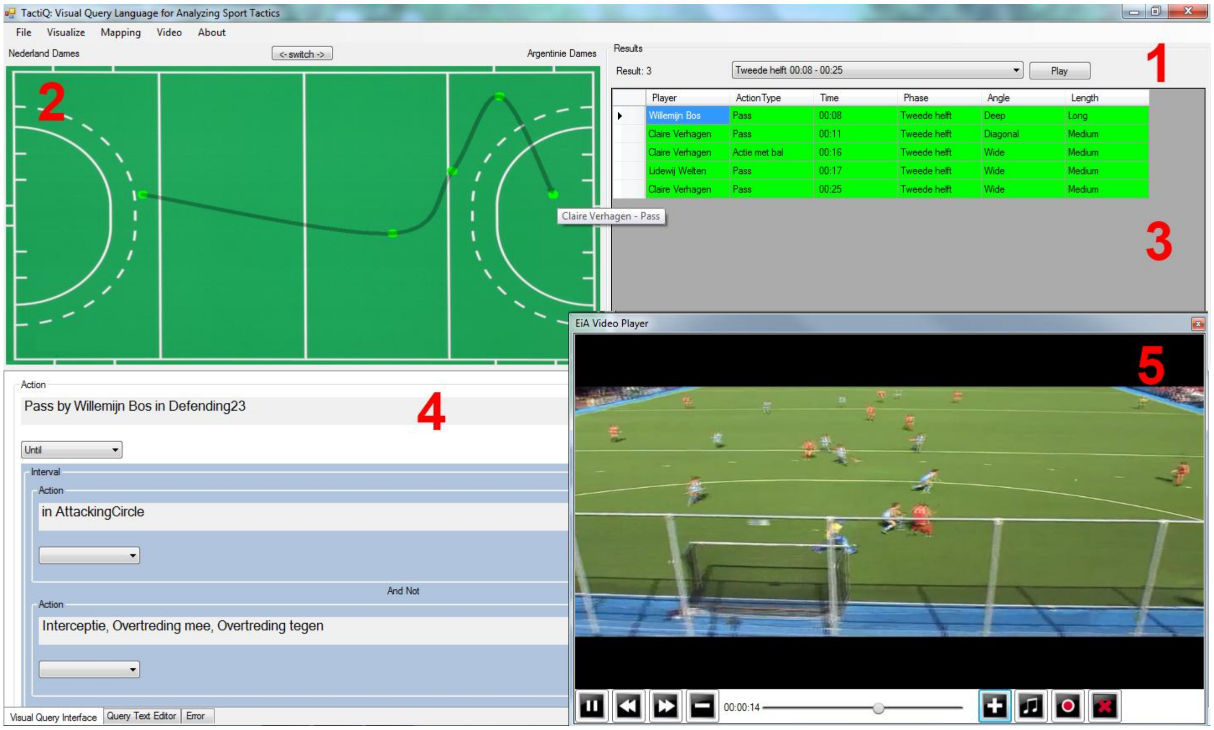

Figure 15: Overview user interface

7.2 Query construction

With the DSL discussed in chapter 5 a query language is created that has limited and focused expressiveness which is easier to be used and learned by a coach. By hiding Boolean operators the complexity of the language is already reduced. But still the language has to be learned especially for people without programming experience. This is not a desirable situation. Therefore, visual query interfaces for constructing a query is created were this DSL syntax is hidden.

The DSL is a fluent interface inspired by natural language. Properties of an action are modeled like adjectives in the language. In the DSL this is done by method chaining which produces a sentence describing an action. But still some language specific syntaxes are remaining in the language. When looking at the following example query in the DSL:

```
all().ByTeam(NederlandDames).Type(Pass).Length(Long)
```

The brackets and the point to mark the beginning of a new method are required. In the visual query interface these constructions are hidden. The same query is represented in the visual query interface in the following way:

Long Pass By Nederland Dames

The visual query interface supports adding specifications of an action by selecting the type of specification (Team-Specification, Player-Specification etc) and then providing the possible values (the team: NederlandDames). The whole sentence can be constructed by a few mouse clicks. To explain the flow of creating a tactical query with the user interface the following use case is constructed: "*Find the pass resulting in a goal attemp*". The steps are visualized in Figure 16.

1. Right click
2. Select Type to define the type of action (1)
3. Select one or more action types (2)
4. Query string is created (3), repeat step 1-4 to define the team
5. Select temporal operator FollowedBy (4)
6. System creates a second action (5),
7. repeat step 1 to 5 for the second action

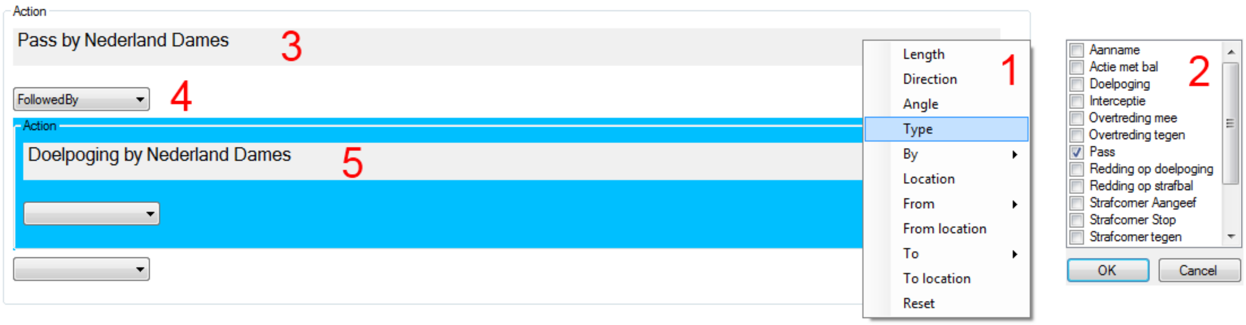

Figure 16: Visual query construction

The user interface is extendable and more operators (both boolean and temporal) can be chained together to create more complex queries.

A query created in the visual editor is converted to a representation in the DSL which can be viewed by the coach in the textual query editor. A coach also has the possibility to create a query directly in the textual editor. In the textual editor a simple for of syntax highlighting is implemented. The query discussed in the previous paragraph with syntax highlighting is:

```
all().Type(Pass).ByTeam(NederlandDames).FollowedBy(all().Type(Doelpoging).ByTeam(Nede
rlandDames))
```

7.3 Result visualization

An important step in the system is the result representation. The results provide the information that a coach uses to gain a better insight in the tactical situations. Results are presented in different ways with different purposes. The following result representations are implemented:

1. Tabular
   A tabular result representation shows in a table the actions that are in a certain result (interval). The actions are sorted by time. The action shows the data in the domain model. The tabular respresentation provides detailed insight in the actions like the players active, type of actions etc.

2. Video
   One of the requirements formed in chapter 3 is that a coach wants to look back a situation in the match. With the query system a moment can be found and with linking it to video file a coach can look the moment back. This gives extra insight in aspects that are not in the match annotation data. For example the match annotation contains only information about the players with the ball, so movements by players without the ball cannot be derived. Using the video a coach can gain better insight in the off the ball actions of the players. The video element is implemented in a separate window so it can be shown on an external monitor.

3. As mentioned in chapter 2 spatial data is inherently visual. Therefore, a spatial visualization component is implemented. Using a flow map also the temporal component is visualized. A flow map shows the movement of object in a spatial domain [28]. It provides insight in the movement of the play over time and can be used to evaluate the patterns of play. An example can be found in Figure 15 in the box with number 2.

4. For the visualization of the temporal compontent of a result a timeline is used. The timeline shows the relation between a result of a query and its occurance in time [31]. It shows how it relates to the time in the match but also provides insight on the length of a result. When the match evolves a certain tactical situation can occur more often for multiple reasons. Players can get tired for example. An example of the timeline can be found in Figure 17 for a match with extra time.

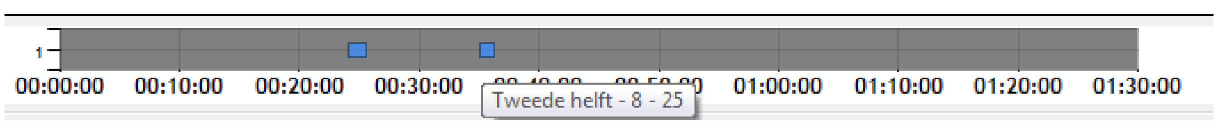

Figure 17: Timeline visualization

7.4 Editing definitions

One of the aspects that supports the flexibility of the system is that a coach can create his own definitions. These definitions relate to the classification of the spatial derivates like the lengths and areas of the field. These definitions are stored in an XML file. The definitions describe the mapping from the match annotation to the domain model. A definition describes a range of values on the level of the match annotation and how it is mapped to a class in the domain model. We want a coach to create a definition but we don't want a coach to reason at the level of the match annotation. Therefore a visual editor is created. With the visual editor the spatial data is visually represented and the match annotation is hidden. By using a visual representation of the spatial data a coach can reason on the domain level instead of the match annotation level. In Figure 18 the editor for the area of the field can be be found. The interface contains multiple regions with different functionalities. The regions are numbered in the image and have the following short explanation:

1. Location overview
   Overview of the current available locations definitions In this region a location can be selected to be edited. Location can also be created and deleted.
2. Location naming
   A location has an unique name or alias which a user can edit.
3. Include areas overview
   A location consist of one or more spatial areas. An area can be a rectangle or a circle. In this region a spatial area can be selected for editing, new areas can be created and existing areas can be deleted.
4. Exclude areas overview
   When desribing the defending 23 zone, the spatial areas desribing the defending circle have to be excluded
5. Visual representation of a spatial area
   A spatial area is visualized on the field. The area can be edited using the mouse and clicking on the field te define an area. New areas are created in a similar way.

Figure 18: Editing spatial location definitions

8 Evaluation

In order to validate the steps taken in the previous chapters an evaluation of the prototype is performed. In the evaluation the developed prototype is tested and data is gathered to determine if the developed system answers the original research question. During the evaluation process several aspects about the prototype are measured. This chapter has the following structure: in the first section the methodology of the evaluation will be discussed. The second section describes the result of the evaluation. In the third section the results of the evaluation are analyzed and a conclusion is drawn.

8.1 Methodology

This section covers the evaluation methodology of the prototype. To perform the evaluation a user-test is performed. This thesis is conducted in a specific domain, field hockey, with a specific target group, the coaches. But a problem arises with the evaluation because the target group is small. Match evaluation via a computer is only applied in the top competitions of field hockey. Therefore, the number of coaches who can participate in the evaluation is limited. This has influence on the evaluation, because of the small target group it is hard to draw general conclusions from the results and the results can only be seen as indicators. On the other hand the coaches working in the top competition can be seen as experts in the domain of field hockey and their feedback and opinion is of great value. In the evaluation four coaches participated which are active on a high national and international level.

During the evaluation is evaluated if the system matches its original goal and its main requirements. The goal of this thesis is providing insight in sport tactics. One of the requirements of the system is the *expressiveness*. A coach must be able to create a tactical pattern with concepts from the domain of field hockey. The second requirement is the *flexibility*. A coach must be able to incorporate his own view of the game in the system and be able define every situation he wants.

The first method in the user evaluation is the USE questionnaire [2]. The questionnaire uses a 7 point likert scale ranging from totally disagree to totally agree in a questionnaire of 30 questions to measure usability based on different criteria. The USE questionnaire measures the following criteria:

- Usefulness
- Ease of Use
    - Flexibility is measured as a subset of ease of use
- Ease of Learning
- Satisfaction

The questionnaire can be found in appendix B. The usefulness is used as a measure to determine if the system reaches its original goal. The ease of use and ease of learning are indicators that the expressiveness and the domain concepts are understandable for the coach.

Next to the questionnaire, a user-test is conducted. The developed system can be seen as a query system, it enables a coach to query a tactical pattern. The evaluation method is based on an evaluation method for query languages. Reisner describes the following tasks that can be performed during a user evaluation of a query language:

- Query writing: A user creates a query based on a question in natural language.
- Query reading: A user gets a query and is asked to reverse engineer to original question in natural language. [29]

With these tasks the complexity of the language constructions which are implemented to describe a tactical pattern are evaluated. The constructions that are evaluated are:

- Adjectives / Method chaining (adding specifications of a action)
- Conjunction (using a dash)
- Disjunction (using a comma)
- Temporal operators

During the user test, measurements can be taken which include the correctness and time to perform a task. In this thesis, the time to perform a task is irrelevant. Time can be used to benchmark the performance against another query system, but there is no other system available to create a tactical query in field hockey. The correctness is an indication if a coach is able to create and understands the expression of a tactical pattern.

During the evaluation the coaches that participate in the evaluation first get a brief introduction in the system. This gives them the same basic knowledge of the system and ensures that they all have the same level of knowledge. The individual evaluation program takes about 1 hour and consists of the following steps:

- Introduction in the dataset
- Short training in the prototype
- Query reading
- Query writing
- Questionnaire

8.2 Results of user evaluation

In this section first the results from the evaluation process are laid out. In Table 4 the result for of the USE questionnaire are summarized. The detailed data of the USE questionnaire can be found in appendix B Looking at the data from the USE questionnaire the prototype scores a high grade (likert scale 1 to 7) on all the categories. Especially the flexibility criterion receives high grades.

Table 5: USE questionnaire result

Measurement	Value
Usefulness	5,88
Ease of Use	5,68
Flexibility	6,5
Ease of Learning	5,94
Satisfaction	6,21

The evaluation of the language constructions of the query construction process can be found in Table 5. The details of the evaluation are laid out in appendix C .The coaches were able to create queries which contained no temporal operators without any problems. Applying conjunction and disjunction structures using the language construct caused no problems. When the queries were getting more complex because of the temporal operators or nesting the coaches had more problems to create a correct query. But for the coaches it was the first time they worked with the systems and they had no knowledge of temporal reasoning. Some of the coaches had a few problems with an ambiguity in the system.

Some queries can be modeled using different approaches and get the same result. An example can be found in the following query:

Give the passes from playerA to playerB

This query can be modeled in the following two ways which express the situation and produce the same result:

- `all().Type(Pass).ByPlayer(PlayerA).ToPlayer(PlayerB)`
- `all().Type(Pass).ByPlayer(PlayerA).FollowedBy(all().ByPlayer(PlayerB))`

But the coaches stated that they understood the ambiguity after a short explanation and stated that when they worked more often with system this wouldn't be a problem.

Another problem was the naming which causes confusing with the *From* and *In* by the location of an action. A *From* location relates to the location of the previous action while *In* location relates to the location of the current action. But the word *From* was interpreted by some of the coaches as the location of the current action.

Table 6: Query language evaluation result

Construction	Score	Remarks
Adjectives	+	Works good, no problems
Conjunction	+	Works good, no problems
Disjunction	+	Works good, no problems
FollowedBy	+/-	Little confusion with from/to in DSL vs followedBy/precededBy
PrecededBy	+/-	Little confusion with from/to in DSL vs followedBy/precededBy
Since	+/-	1 contestant had problems with the English naming, contestants were able to create an interval query almost correct in 1 turn
Until	+/-	1 contestant had problems with the English naming, contestants were able to create an interval query almost correct in 1 turn

8.3 Conclusion

This section is dedicated to the analyses of the results of the evaluation and based on this analyses a conclusion is drawn. In the introduction of this chapter the requirements of the system were briefly mentioned. This section discusses every requirement based on the results of the user evaluation and the design choices during the whole process.

Usefulness

The system was developed to provide insight for a coach in tactical situation. With this requirement is measured if the system fulfills this goal. Looking at the results from the USE questionnaire the results of the usefulness of the system were positive. One of the coaches even stated that "*It is the next step in match evaluation".*

Expressiveness

Looking at the results from the USE questionnaire the coaches involved in the evaluation considered the system to be user friendly which is an indicator that the expression of tactical pattern is at an understandable abstraction. But from the query reading and writing evaluation some issues arise. The language has an ambiguity which was confusing for some of the coaches. Furthermore, some coaches found the naming of *From* and *In* with locations confusing. But after a short explanation the coaches understood the language. The perceived learnability of the language was rated high by the coaches according to the results from the USE questionnaire. Looking at the evaluation process, coaches were able to create queries after a short training. This aspect can also be considered as an indicator that the expression is at the abstraction level of the coach.

Flexibility

From the USE questionnaire coaches stated that the system was flexible. Coaches had the idea it could express the tactical situation they wanted insight in. A remark has to be made about this statement. The coaches involved in the evaluation were new to data analysis and the system. It is hard for them to judge if the system is flexible enough to express all the tactical questions they might have in the future.

9 Conclusions and future work

Based on the data gathered in the evaluation this chapter answers the research question of this thesis and gives an overview of the contributions of this thesis work and their relation to the result of this thesis. The second section gives an overview of future work.

9.1 Conclusion

The goal of this work is to enable a coach to define and lookup tactical patterns to provide a better insight in the tactical performance of a team. A tactical pattern relates to the spatial-temporal aspect of the match annotation data. To declare a tactical pattern a coach must express the spatial-temporal relations of the match annotation data. In this thesis a system is designed and a prototype of the system is implemented. To prototype is evaluated to answer the following research question:

How to provide insight in sports tactics to a coach via a query system in a user friendly and flexible manner using a historical spatial-temporal dataset?

Looking at the results of the evaluation the system is providing insight in tactical situations. Coaches were able to define a tactical situation and evaluate the result. The prototype scored high on the usefulness criteria in the user evaluation. The coaches involved in the evaluation perceived the prototype user friendly and easy to learn and were able to create a query after a short introduction despite that some aspect in the language can cause confusion because of the naming or the ambiguity. The coaches perceived the system to be flexible.

A remark on these results is that because of the small group of coaches that participated in the evaluation it is difficult to draw conclusion from the evaluation although the group of coaches can be considered experts. The results should therefore be interpreted as indicators.

In the first chapter a number of objectives where set. These objectives contributed to the overall outcome result of this work. Each objective contributed a specific part and combined they form the fundaments of this thesis work.

Requirement analyses

Via a domain study a number of representative tactical question were gathered. These questions were analyzed and provided the components necessary to express a tactical question. The domain study also resulted in the functional requirements of a software tool to analyze tactics.

Domain model

In the data model the data is modeled at a higher abstraction. Both the spatial and temporal relations are made explicit which offers a richer expression of the data. The domain model enables the expression of a tactical pattern based on the requirements analysis. The data model aids the flexibility criteria by making it configurable to the view of the coach.

Query language

As a query language a domain specific language is designed and implemented. This DSL enables the expression of a single action and chaining action together to create a sequence of actions using temporal operators. The DSL offers expressiveness limited to the creation of a tactical pattern. To ease the communication with the coach Boolean constructions are hidden.

User Interface

The GUI enables a coach to create a tactical pattern using mouse operations and hides the syntax of the DSL which offers a easy to use access to the system. It offers multiple ways of result visualizations which are based on the requirements analyses to cover the different aspect of the results. To configure the system to the view of the coach the system offers a visual editor to edit the definitions.

9.2 Future work

In the current system it is only possible to view 1 result of a tactical query at the time. It can be interesting to aggregate these results. An example is a flowmap were the spatial paths are clustered into the dominants paths [28]. Using a flowmap where the paths are clustered a coach can discover patterns of play. The clustering can also be applied on the data. For a certain tactical situation like the creation of goal attempts it can be interesting to know which players are involved and how many times. Other interesting cases are the average duration of moments. To enable such operations the DSL can be extended with mathematical operations like *GroupBy*, *Count, Average* etc. These operations will increase the expressiveness of the language but it can be challenge to make them accessible in a user friendly manner.

In match annotation the time is modeled as an event and with a tactical query it can be converted to an interval. An event is a temporal entity with an occurrence in time and an interval is temporal entity with a beginning in time and an ending in time [36]. For reasoning with interval entities different operators exist. Examples of operators for interval time are: *Before, Equal,Meet, During, Cross, Finish, Start, After, Inside* [5] With these operators can be derived which player initialized a tactical situation of example. Combining the aggregating of the result discussed in the previous paragraph with these temporal operators more insights can be provided in patterns of play.

The user interface in the current system is a projection of the linguistic elements of the DSL. It enables a user to create a tactical query by a few mouse clicks. A query is formed as a sentence in natural language. This approach still requires some knowledge of temporal constructions and for long queries it can be difficult to express the query in a linguistic sentence. In other work [14][18] attempts can be found to visualize temporal relations, combining these visual techniques with the approach of this work can be interesting to create a user friendlier system.

In the mapping of the match annotation data to the domain model a crisp classification is used. The length of an action is for example short or long. For example a length smaller than 4 meters can be classified as short and a length greater than 4 meters as long. But the values 3.9 meters and 4.1 meters differ little but are classified in different classes. A solution for this problem can be found in fuzzy logic. Fuzzy logic attempts to model human sense of words and uses a degree of membership instead of crisp boundaries [26]. Using linguistic variables a length can be identified as *very short* and *slightly long* for example. Applying fuzzy logic the data can be modeled at a level closer to human reasoning which can increase the usability. Applying fuzzy logic causes that a query gets more results and a sorting based on a measurement of relevance is desirable.

In the current system the status of the match like the current score of a match or the player locations in the field are not modeled. This information is currently not available in the match annotation set but can be of great value. The current score for example can have influence on how a team acts in certain tactical situation. When not in lead a team can adopt a more offensive way of playing for example. The locations of the players without the ball and there movement gives insight in the behavior of a player without the ball and is an aspect of the tactical performance in field hockey [10].

The EIA annotation methodology tries to describe the fundamental elements of field hockey but during the interviews with the coaches some aspect coaches want to see are currently not annotated. An example can be found in the goal attempts. In the current methodology a goal attempt handling cannot be split based on the techniques used. A goal attempt can be a hit with forehand or with backhand, a tip/deflection. In the EIA methodology the subdivision of goal attempts is supported by adding an extra annotation to an action but this is currently not applied in EIA field hockey. More domain research is needed to describe all the fundamental elements in field hockey.

Future research can also be found in the performance evaluation of tactics. With this thesis insight can be gathered about the tactical behavior of a team in certain situations. Using the clustering of tactical situation and applying statistics the relation between tactical patterns and performance can be investigated. This could lead to a list of general patterns of play. An example can be found in tennis: The United States Tennis Association recommends 58 winning tactical patterns of play in specific situations [34]. When more data becomes available all the matches of the last 5 years for example trends and development of the game can also be investigated.

In the current system an InMemory strategy is used; all the objects are loaded mapped to the domain model and loaded into the memory. When evaluation multiple matches this solution might not be feasible. Furthermore, when executing a query containing a temporal operator every action can be evaluated on every other in a phase of the match. The temporal operator has a quadratic complexity. Future research can be conducted in the Query Optimization of spatial-temporal queries to find effective strategies to execute such queries.

References

[1] Abiteboul, S. et al. 1996. Temporal versus first-order logic to query temporal databases. *Proceedings of the fifteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems* (Montreal, Quebec, Canada, 1996), 49–57.

[2] Arnold M. Lund 2001. Measuring Usability with the USE Questionnaire. *STC Usability SIG Newsletter*. 8, 2 (2001).

[3] Carling, C. et al. 2006. *Handbook of Soccer Match Analysis: A Systematic Approach to Improving Performance*. Routledge.

[4] Carling, C. et al. 2009. *Performance Assessment for Field Sports*. Taylor & Francis.

[5] Cavalcanti, V.M.B. et al. 2006. Querying spatio-temporal databases using a visual environment. *Proceedings of the working conference on Advanced visual interfaces* (Venezia, Italy, 2006), 412–419.

[6] Chang, S.-K. et al. 2004. Querying Distributed Multimedia Databases and Data Sources for Sensor Data Fusion. *IEEE Transactions on Multimedia*. 6, 5 (Oct. 2004), 687–702.

[7] Chomicki, J. Temporal query languages: A survey. *Temporal Logic*. D.M. Gabbay and H.J. Ohlbach, eds. Springer-Verlag. 506–534.

[8] Deursen, A. van et al. 2000. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.* 35, 6 (2000), 26–36.

[9] Egenhofer, M.J. 1994. Spatial SQL: A Query and Presentation Language. *IEEE Trans. on Knowl. and Data Eng.* 6, 1 (1994), 86–95.

[10] Elferink-Gemser, M.T., Visscher, C., Lemmink, K.A.P.M., Mulder, T.W. 2004. Relation between multidimensional performance characteristics and level of performance in talented youth field hockey players. *Journal of Sports Sciences*. 22, 11-12 (2004).

[11] El-Mahgary, S. and Soisalon-Soininen, E. 2007. A Two-Phased Visual Query Interface for Relational Databases. *Proceedings of the 18th international conference on Database and Expert Systems Applications* (Regensburg, Germany, 2007), 489–498.

[12] Evans, E. 2004. *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley.

[13] Evans, E and Fowler, M Specifications.

[14] Fernandes, S. et al. 1997. Visual query operators for temporal databases. *Temporal Representation and Reasoning, 1997. (TIME '97), Proceedings., Fourth International Workshop on* (May. 1997), 46–53.

[15] Fister Jr, I. et al. 2012. Implementation of the Domain-Specific Language EasyTime using a LISA Compiler Generator. *arXiv:1206.2542*. (Jun. 2012).

[16] Fowler, M. and Parsons, R. 2010. *Domain-specific languages*. Addison-Wesley.

[17] Gabbay, D.M. and McBrien, P. 1991. Temporal Logic & Historical Databases. *Proceedings of the 17th International Conference on Very Large Data Bases* (1991), 423–430.

[18] Hibino, S. and Rundensteiner, E.A. 1997. User interface evaluation of a direct manipulation temporal visual query language. *Proceedings of the fifth ACM international conference on Multimedia* (Seattle, Washington, United States, 1997), 99–107.

[19] Huo, J. 2010. *A formalism for visual query interface design*. UNIVERSITY OF WATERLOO.

[20] Jan Chomicki and David Toman 2009. Temporal Logic in Database Query Languages. *Encyclopedia of Database Systems*. Springer. 2987–2991.
[21] Jaymin Patel 2003. *Temporal Database System*. Dept of Computing, Imperial College.
[22] Kuijpers, M.J. 2010. *Supporting the Tennis Coach: Automatically Analyzing and Evaluating Tennis Footage*. Web Information Systems, Department of Software Technology, Faculty EEMCS, Delft University of Technology.
[23] Lewis, M. 2005. *Moneyball: The Art of Winning an Unfair Game*. W. W. Norton & Company.
[24] Li, X. and Chang, S.K. 2004. An interactive visual query interface on spatial/ temporal data. *in: Proceedings of the 10th International Conference on Distributed Multimedia Systems* (2004), 257–262.
[25] Mueller, J. 2010. *Professional IronPython*. Wiley Pub.
[26] Negnevitsky, M. 2005. *Artificial intelligence : a guide to intelligent systems*. Addison-Wesley.
[27] Pelekis, N. et al. 2004. Literature review of spatio-temporal database models. *Knowl. Eng. Rev.* 19, 3 (2004), 235–274.
[28] Phan, D. et al. 2005. Flow Map Layout. (2005), 29.
[29] Reisner, P. 1981. Human Factors Studies of Database Query Languages: A Survey and Assessment. *ACM Comput. Surv.* 13, 1 (1981), 13–31.
[30] Schumaker, R.P. et al. 2010. *Sports Data Mining*. Springer.
[31] Silva, S.F. and Catarci, T. 2000. Visualization of Linear Time-Oriented Data: A Survey. *Proceedings of the First International Conference on Web Information Systems Engineering (WISE'00)-Volume 1 - Volume 1* (2000), 310.
[32] Stef Klein Koerkamp 2004. *Effectiviteit in Actie*. Faculteit der Economische Wetenschappen, Vakgroep Econometrie en Operationele Research, Rijksunivesiteit Groningen.
[33] Stephen John Williams and Lawrence R. Kendall 2007. A profile of sports science research (1983—2003). *Journal of Science and Medicine in Sport*. 10 (2007), 193—200.
[34] United States Tennis Association 1996. *Tennis tactics : Winning patterns of play*. Human Kinetics.
[35] FIH - Rules of Hockey 2012.
[36] 2011. Temporal Databases and Mediators. *Temporal Data Mining*. Chapman and Hall/CRC. 1–20.

Appendix A: **Effectivity in Action**

EIA is the match analysis notation methodology of ORTEC originally designed to measure the effectiveness of players during a match. The notation design is afterwards extended with spatial information to gain more insight in the spatial component of game. This appendix has the following structure: first the notation or what is saved will be discussed followed by the procedure.

A. Notation

With Effectivity in Action (EIA) notation actions are tagged during a sport match analysis. This can be a live analysis or an after match analysis from video. An action is defined in EIA as an action with the ball by a player, or an attempt of a player to play the ball during the match. An example of an action with the ball is a pass by a player, an example of an attempt to play the ball is a failed save on a goal attempt by a goal keeper were the keeper doesn't touch the ball. In EIA actions are graded based on their effectiveness and are therefore called Judgements

A Judgement has the following attributes:

- Match
  - Describing the match where the Judgement has taken place, the teams involved the competition, date etc.
- Category
  - A category describes the type of action, to distinguish action from each other based on type.
- Player
  - Is describing the player who performed the action including information like the name, shirt number, and the position role (Goal Keeper, Defender, Midfielder, and Attacker)

  Team
- Spatial information
  - X and Y location on the field on a scale from 0 to 100, see Figure 1
- Temporal information
  - Phase
  - Time in the phase in seconds
- Grade describing the effectiveness of an action on a scale of 1 to 5

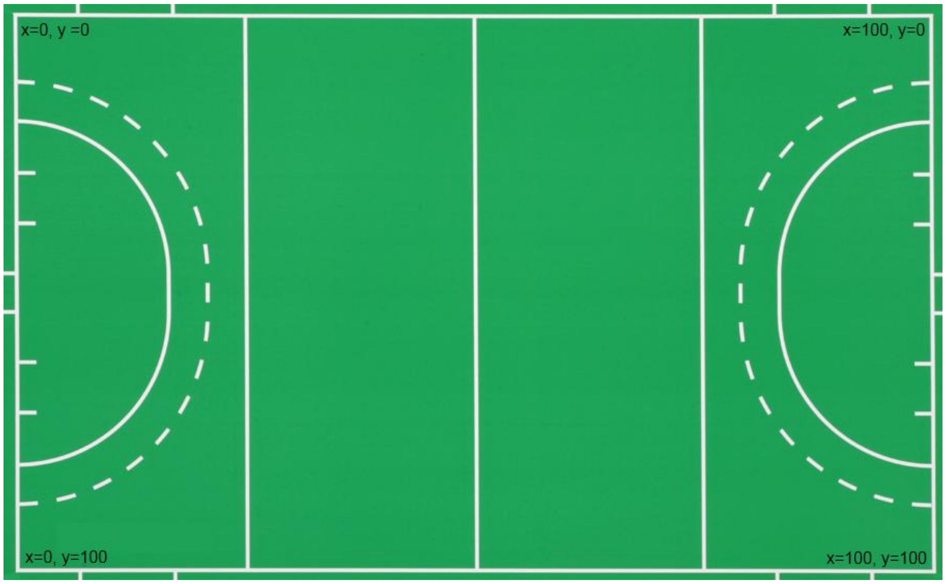

Figure 1 Location definitions

The data is stored in a conventional RDBMS in de normalized structure. The Entity Relation Diagram can be found in figure 2. An example of the data in a denormalized view of the database structure were all the database keys are reviewed showing only the notation can be found in table 1.

Table 1: Notation example

Phase	Time	Team	Shirt#	FirstName	SurName	Category	LocX	LocY
1	40	Nederland	11	PlayerName	Surname	Defending	52,9	44,9

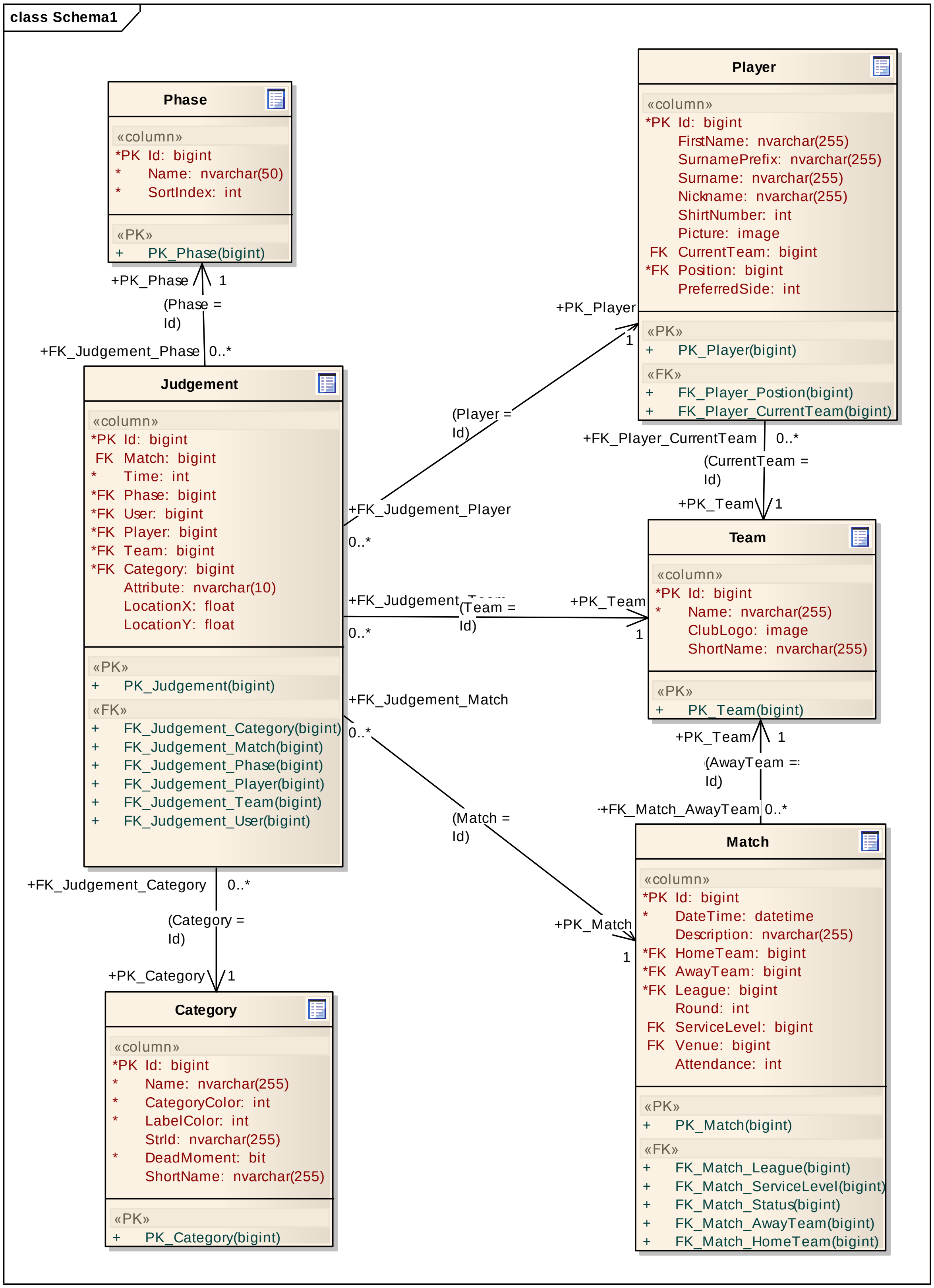

Figure 2 Entity relationship diagram EIA

The actions are annotated by a type of action and a grade. The action types are sport depended. For field hockey the following action types are defined:

Table 2: Action types

Pass	Save on goal attempt
Interception	Save on penalty corner
Attacking action	Save on penalty shot
Defending action	Penalty corner shot
Receiving	Penalty corner pass
Goal Attempt	Penalty corner stop
Penalty shot	Pass
Dribble	Interception
Free hit	Free hit self pass

B. Process of notation

A match is analyzed by four analysts, two for each team. For each team we can split the two analysts in two different roles, a caller and a processor. The caller is watching the match and calling the action; the caller is recognizing the player, the type of action and assigns a grade for the effectiveness of the action. While watching and calling the analyst is also responsible for the spatial locations. When an action occurs the location is logged by the caller using a tablet computer, the caller clicks on a field to specify the location of the action. The moment of entering the location is the moment that the time in the game for the action is determined. The location is relative to the team they are analyzing based on the direction of play. The spatial orientations of both teams differ. The location of the own goal is located near the x = 0 location. When an action takes place involving both teams the locations entered by both teams are each other complement. The actions called by the caller are processed by the second analyst the processor. The processor listens to the caller and assigns the action types, players and grade to the locations created by the caller. During a match analysis either teams can be analyst simultaneously or a match is analyzed twice, for each team ones. The EIA notation is centrally saved on a server. A global overview of the EIA notation system can be seen in Figure 3

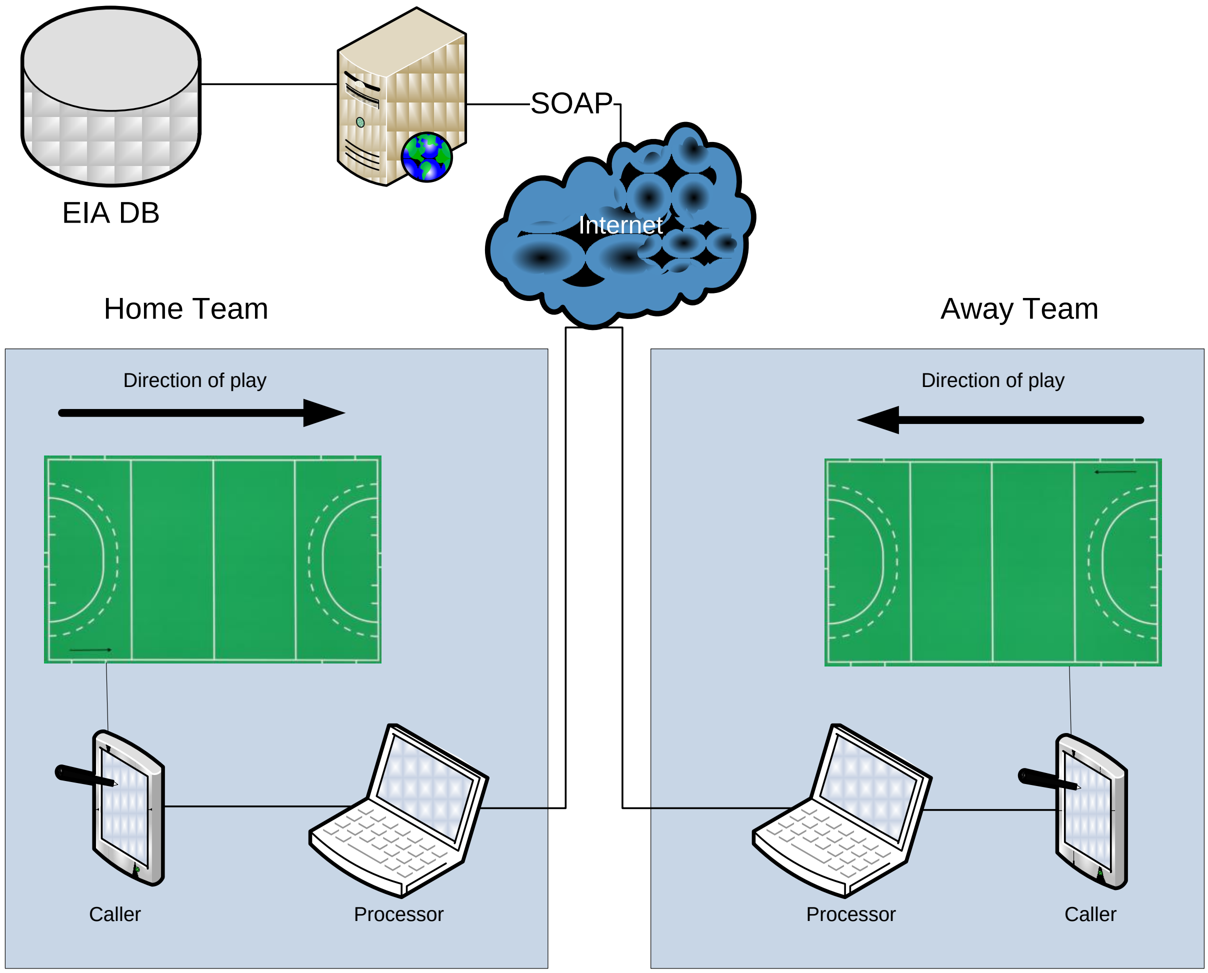

Figure 3 Global overview EIA Notation

Appendix B: USE questionnaire results

Usefulness		5,88	
1	It helps me be more effective.		6,5
2	It helps me be more productive.		6,25
3	It is useful.		6,5
4	It gives me more control over the activities in my life.		5
5	It makes the things I want to accomplish easier to get done.		6,25
6	It saves me time when I use it.		4,25
7	It meets my needs.		6,25
8	It does everything I would expect it to do.		6
Ease of Use		5,68	
9	It is easy to use.		6
10	It is simple to use.		6,25
11	It is user friendly.		6
12	It requires the fewest steps possible to accomplish what I want to do with it		5,75
13	It is flexible.		6,5
14	Using it is effortless.		5
15	I can use it without written instructions.		4,5
16	I don't notice any inconsistencies as I use it.		5
17	Both occasional and regular users would like it.		5
18	I can recover from mistakes quickly and easily.		6,5
19	I can use it successfully every time.		6
Ease of Learning		5,94	
20	I learned to use it quickly.		6,5
21	I easily remember how to use it.		5,75
22	It is easy to learn to use it.		5,75
23	I quickly became skillful with it.		5,75
Satisfaction		6,21	
24	I am satisfied with it.		6,25
25	I would recommend it to a friend.		6,5
26	It is fun to use.		6,25
27	It works the way I want it to work.		5,75
28	It is wonderful.		5,75
29	I feel I need to have it.		6,5
30	It is pleasant to use.		6,5

Appendix C: **Language evaluation**

The evaluation consists of two tasks: query reading and query writing. During these tasks the correctness is measured. With these tasks the following language construction are evaluated

- Adding properties like adjectives
- Conjunction
- Disjunction
- Temporal
    - FollowedBy
    - PrecededBy
    - Since
    - Until

To evaluate the language questions are created. The questions are classified which construction is applied. The classification can be found in table 1.

Table 1

Reading		Adjectives	Conjunction	Disjunction	FollowedBy	PrecededBy	Since	Until
	1	x						
	2	x						
	3	x	x					
	4	x					x	
	5	x						x
	6	x			x			
Writing	1	x						
	2	x		x				
	3	x						
	4	x					x	
	5	x			x			
	6	x				x		

A. Query reading

1. `all().Type(Doelpoging)`
    a. Show all goal attempts
2. `all().ByTeam(NederlandDames).Type(Pass).InLocation(DefendingMidfield).ToLocation(AttackingMidfield)`
    a. All passes from defending midfield to attacking midfield

3. `all().ByTeam(NederlandDames).Type(Interceptie).InLocation(AttackingMidfield-Center)`
    a. All interception action in attacking midfield
4. `all().ByTeam(NederlandDames).Type(Doelpoging).Since(all().InLocation(Defending23, DefendingMidfield), all().Type(Interceptie, Verdediging))`
    a. Outlet leading to goal attempt
5. `all().ByTeam(NederlandDames).Type(Pass, Vrijslagselfpass).InLocation(Defending23).Until(all().InLocation(AttackingMidfield-Right), all().Type(Interceptie, Verdediging))`
    a. Outlet via right zone of the field
6. `all().ByTeam(NederlandDames).ByPlayer(Bos).FollowedBy(all().ByPlayer(deGoede).Until(all().InLocation(AttackingMidfield), all().Type(Interceptie, Verdediging)))`
    a. Play from Bos to deGoede resulting in Attacking23 zone

B. Query Writing

1. All goal attempts NederlandsDames
    a. `all().Type(Doelpoging).ByTeam(NederlandDames)`
2. Alle short passes from Agliotti to Lammers
    a. `all().ByPlayer(Agliotti).Length(Short).Type(Pass).ToPlayer(Lammers)`
3. Alle circle penetraties
    a. `all().ByTeam(NederlandDames).Type(Actiemetbal, Pass).ToLocation(Attacking23)`
4. Cause of a goal attempt from, outlet from own half
    a. `all().ByTeam(NederlandDames).Type(Doelpoging).Since(all().InLocation(Defending23, DefendingMidfield), all().Type(Interceptie, Verdediging))`
5. Show all Possession loss
    a. `all().ByTeam(NederlandDames).FollowedBy(all().Type(Interceptie, Verdediging))`
6. How is a penalty corner created?
    a. `all().Type(Strafcornervoor).PrecededBy(all().ByTeam(NederlandDames))`

C. Results

During the evaluation the people the results of the queries where analyzed on the correctness and there values are mapped to the different language constructions. Table 2 shows the result per language construction using a 3 point likert-scale.

Table 2

Construction	Score	Remarks
Adjectives	+	Works good, no problems
Conjunction	+	Works good, no problems
Disjunction	+	Works good, no problems
FollowedBy	+/-	Little confusion with from/to in datamodel. fromPlayer, toLocation
PrecededBy	+/-	Little confusion with from/to in datamodel. fromPlayer, toLocation
Since	+/-	1 contestant had problems with the English naming, contestant picked up temporal reasoning quite good
Until	+/-	1 contestant had problems with the English naming, contestant picked up temporal reasoning quite good