Eindhoven University of Technology
Department of Mathematics and Computer Science
PDEng Mechatronics System Design 2021 - 2023

Architecture Description for Autonomous Referee System

The architecture description report for the block 2 in-house project.

Authored by the MSD team 2021-2023 under the guidance of Robert Deckers.

Supervisors:
Rene van de Molengraft
Erjen Lefeber
Robert Deckers

Version 4.0

Friday 25th March, 2022
Eindhoven

Abstract

The document is the architecture description of the AutoRef project of the MSD 2021-2023 group. The problem is introduced in chapter 1, in chapter 2 we analyse different stakeholders and their concerns. In Chapter 3 the neighbouring systems, their interactions and relationships with the AutoRef is discussed. Finally, different architecture models are shown from different viewpoints, which address the stakeholder concerns.

Appendix contains a glossary, open design questions for the future teams to be answered and additional information on functional decomposition.

Intended Audience

The future teams should consider this as a reference and could consider the information and recommendation presented here as a starting point for their work. It is recommended that the project manager and the system architect read this thoroughly. The designers could read Future Work section 3.7 and 4.1.2 to get started with the project immediately. It could also help the designers who has interest in Systems Thinking and System/ Software Architecture to read this document thoroughly.

Table of Contents

1. Introduction

This document contains the architecture description for the Autonomous Referee System. It identifies different stakeholders interested in this project, lists their concerns, provides rationale for prioritising the concerns, and finally addresses their concerns with concept design solutions.

The project has a legacy of 8 years, during which the past generations of trainees from Mechatronics System Design have contributed in various ways to the design of this system. The main idea here is to build on the previous work and design a platform, which could be a base for the future trainees to contribute to Autonomous Referee for a Robot Soccer match.

This chapter discusses the Goal, Proposed Scope and important discussions on the Application, Design and Realisation of the project and the chapter concludes by suggesting future work.

1.1 Goal from project kick-off meeting

The following words were copied from the project-kick off document handed over to us by Erjen.

The final goal is to have a 5 minute long 2 against 2 robot soccer game, using the Tech United turtles, refereed by the system described above, which receives a positive recommendation by an experienced human referee. By positive recommendation we mean that the human referee acknowledges that the provided refereeing system, for that 5 minute long game, refereed the match well[1].

1.2 Proposed Scope

The goal to be achieved is too ambitious for the project period of 6 weeks. Hence, we as a team, propose a scope that is of interest to our learning goals, which could be achieved in the available time and which will contribute to the above goal.

Tech United[2] is one of the most reputed teams to participate in the RoboCup[3], winning the RoboCup at least a several times. The overall idea of the project is to,understand what could be done with the available time and resources and establish a common platform which could be used to contribute to the development of the AutoRef(Autonomous Referee System). The scope defines what we as a team intend to work on and later propose on how this platform could be extended in different ways in the future to achieve the bigger goal of having an AutoRef.

The idea to have one or more Turtles see figure 1.1 (or robot with similar capabilities) in the field to act as an autonomous referee system on a real robot soccer match. In order to achieve this we take small steps as taking the data from robot players and try to make decisions regarding rule violations for a subset of rules that is difficult for human referees to interpret (FIFA LAW 9, 9.1 9.2, 8.3,15.1, 16.1, 17.1, 13.1 and 13.4) and then validate the performance of the AutoRef with a human referee.

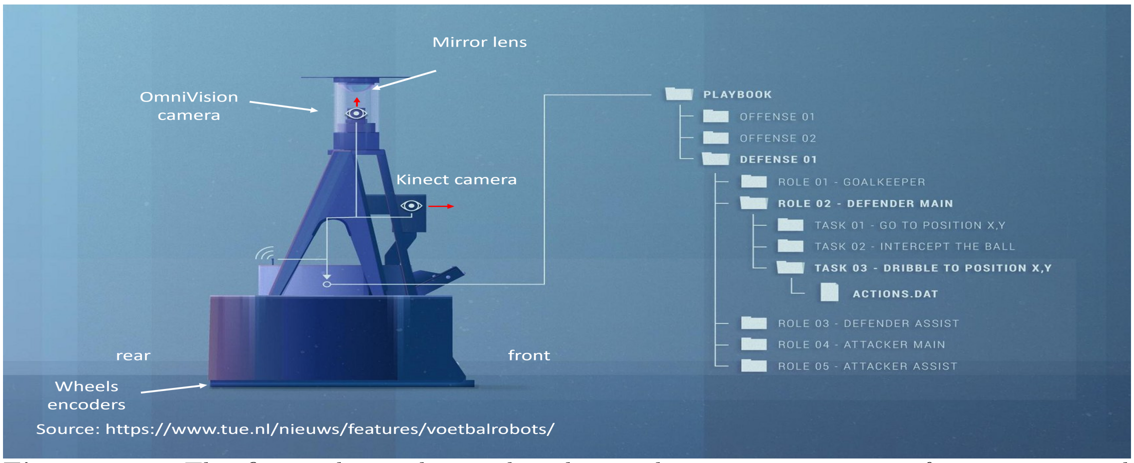

Figure 1.1: The figure shows the turtle robot and gives an overview of its sensors and capabilities[4]

To design an AutoRef, we try to understand how a human referee would tackle this problem. A human referee 1.2 and in some cases, a human referee and a few assistant referees 1.2, 1.3 who have a very good understanding of the rules, run along the soccer field, following the ball and observing the players in the region of interest, overseeing if the rules of the game are being upheld until the match is completed.

Figure 1.2: The figure shows a team of human referees in a soccer game. The figure also shows decisions being communicated to the soccer team manager[5].

Figure 1.3: The figure shows the assistant referee, who is assisting the main referee, by overseeing the lines of the field [6].

On a very high note, the human referee system is gathering information from different sources and from different perspectives, and then use the knowledge of the rules and the information from perception, to make fair decisions on rules and communicate them to the players and others around the game.

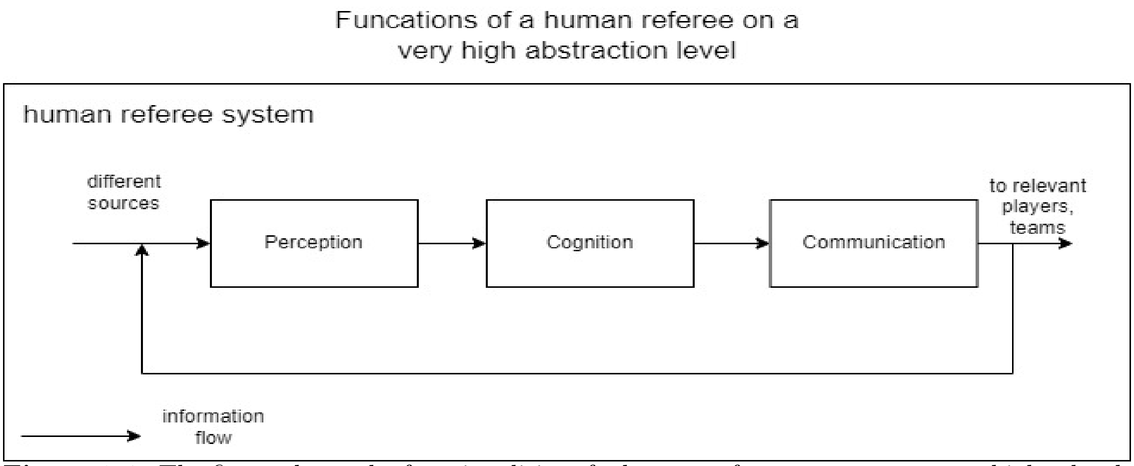

Figure 1.4: The figure shows the functionalities of a human referee system on a very higher level.

It was decided to work on a very high abstraction level by defining a common coding platform, to which the future teams can contribute to.

The AutoRef platform contains 3 functions, the functions were inspired by the functions of a human Referee :

- Perception: Perception refers to the act of gathering data from different perspectives of the game to understand the bigger picture. Perception is a complicated task, we need data from more than one sensor and combine them in a systematic and intelligent way, to capture the complete picture of the game. Several alternatives are explored to investigate the different possibilities. The explored options are documented in feasibility study reports[7, 8, 9, 8, 4].

- Cognition: Cognition forms the brains of the system, here is where the inherent knowledge of the game(rules) is present. Cognition uses the knowledge of the rules and the information received from the perception to make rule violations regarding the game. The maturity and autonomy of the cognition sub-system can be very complicated. *Here, with this project we propose a simple decision maker(a coding platform on Matlab) as a cognitive system for the AutoRef, which over the course of time could be developed to achieve a higher maturity and autonomy.*

- Communication: Communicate the information regarding fouls, rule violations, etc to the human referee or the ref box and the robots involved in the game. The sophistication of the communication system is subjective. *Here, we propose that we only indicate and log the rule violations of the game to be reviewed later for validation by the human referee.*

1.3 Application

This section discusses the application of the project. It discusses the functionalities of the AutoRef and also discusses the usage of the project.

Functionality

The AutoRef is capable of identifying robot positions, ball positions, game states(Kick-Off, Half-Time, Throw-In etc.) from the data received. The AutoRef is capable of detecting if the ball is on the line, inside or outside the line, to detect violation of Fifa laws 9.1, 9.2, 15[1]of the MSL rule-book 2022 V23.0. The AutoRef is also capable of detecting if the ball has rolled at least a minimum of 0.5 m at different game states, to detect violations of Fifa law 8.3, 15.1, 16.1,17.1 of the MSL Rulebook 2022 V23.0. The decomposition of these fifa laws into tasks and skills can be found in the System Requirements in document [10]. These capabilities will be used to detect and report rule violations, corresponding to the respective game state, teams, players, rules and time. The AutoRef is also capable of logging all the data of the game at the time of rule violations.

Usage

The designed and developed AutoRef system will be used as a decision maker to detect rule violations(FIFA LAW 9, 9.1 9.2, 8.3,15.1, 16.1, 17.1, 13.1 and 13.4) [3], based on the data collected from the robot players(TURTLE). The performance of the AutoRef will be validated by the match recording videos and the logged information will be verified and validated by the human referee.

This project work could also be used as a basis for the future groups to build on it.

1.4 Technical Design

This section discusses the design considerations, rationales for not building on previous work and how the proposed scope overcomes the issues of the previous work. This section also discusses the Assumptions made during the design and then speaks about the development and the working of AutoRef.

[1]Fifa law 15.1 to detect the violation is implemented, the procedure after violation is not implemented.

Considerations

The previous group MSD 2020 worked on developing a virtual simulator and implemented rules as indicated in the requirements document [11]. This work could be extended by implementing more rules, we chose not to continue with this work for several reasons:

1. The scenarios of rule violations for the virtual robot soccer match was hard coded. This, hard coding of scenarios does not address third concern of Rene and first concern of Erjen.

2. Virtual data is always reliable, but this is not the case in the ideal world. This does not address third concern of Rene.

3. The extend-ability of the previous project is only restricted to two dimensions, that is implementing new rules or improving the efficiency of the code. This does not address third and fifth concern of Rene.

4. The previous work did not directly contribute the goal of having an Autonomous Referee as there was no hardware involved or the possibilities of interfacing the AutoRef to different hardware options were not discussed. This does not address second concern of Rene and fourth concern of Erjen.

5. The validation of AutoRef in robot soccer match was not carried out. This does not address second concern of Rene and fourth concern of Erjen.

Note: The concerns and the stakeholders are discussed in the next chapter.

The proposed idea of gathering the data from the TURTLEs to make decisions based on the real data, will solve a few of the problems that are listed above. The below points can be considered as answers to the problems mentioned above. For instance, the first point below answers the first point mentioned above.

1. Scenarios were hard coded in the previous AutoRef, we use real data from the robots which participated in a real soccer match and then develop a robust decision making algorithm.

2. The data collected from the robot players(turtles) is uncertain and the developed decision making algorithm will be robust to uncertainties like noise, missing data samples, etc.

3. The extend-ability of the AutoRef is open to many dimensions like implementing more rules, having independent data collection strategy, fusing data from multiple sources, working on bias in the data from the turtles, integrating the AutoRef with the Turtles, improving communication with human referee and AutoRef, better evidence collection and visualisation strategy.

4. The work done will contribute directly to the development of having an Autonomous Referee. By building on this project and making progress on integrating this with turtles can contribute significantly in having an AutoRef.

5. The implemented decision maker will be validated with the video data collected from the static cameras in the lab.

Note: This project is also a collaboration with the Tech United club, which is also **strongly recommended for the future groups**.

Assumptions

1. The data collected from the robot players(turtles) is assumed to be the ground truth, therefore we do not account for the bias in the data.
2. The data is collected using the Tech United code, and we do not hold accountability for the data collection, but just use the data collected from the Tech United code.

AutoRef Development

The AutoRef decision maker is developed in MATLAB. The data is collected by Tech United in the form of .mat files. A modular design approach is followed to give full autonomy to the present designers and the future designers to add more functionalities or to improve the algorithms of the AutoRef. The relevant information from the data is used to make the decisions regarding rule violations. The below figure 1.5 shows the important functions of the AutoRef system. There is only information exchange between the functions.

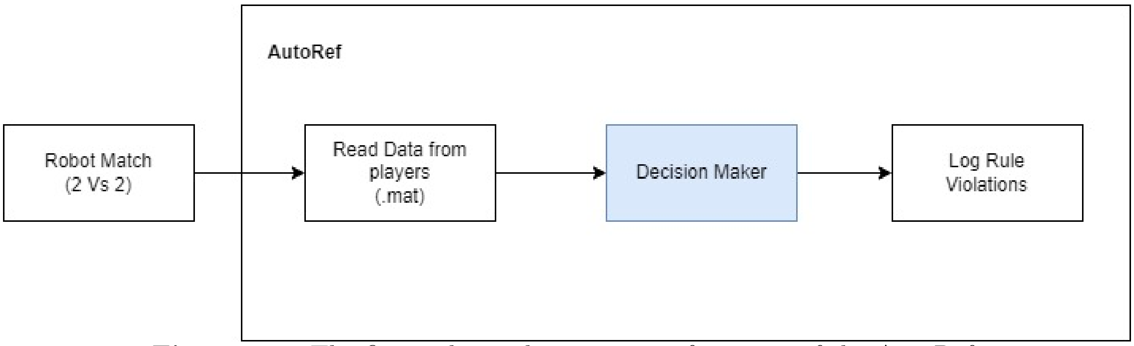

Figure 1.5: The figure shows the important functions of the AutoRef.

AutoRef Working

The AutoRef system takes the data from the actual robot match in the form of .mat files. The data consists of all the information that is logged by the robot players. The information such as ball position, robot position, player orientations etc., is used by the decision maker to make rule violation decisions. The AutoRef works offline, meaning the data collected by the match is then used to make decisions for the rule.

1.5 Realisation

Design Process

The rules which are difficult to interpret for a human referee are discussed with the stakeholders. The stakeholders found rules 8.3, 15.1, 16.1, 17.1 from the rule-book to be most interesting. These are further decomposed into functions. The functions were analysed with Design Structure Matrices(DSM) to develop efficiently. The functions are developed and then unit tested independently. After independently testing the functions, we integrate the functions and validate it with the match videos collected from the stationary cameras.

Verification

The capabilities of the AutoRef are developed as functions in MATLAB and the independent functions are unit tested before integration them for a task.

Validation

The decisions made by the AutoRef are validated in two stages:

- Videos: The decisions made the AutoRef are validated by matching the time-stamps of the video to the time-stamps to where the rule violation occurred. This could also be used to determine the performance of the AutoRef.
- Human Referee: A human referee will validate the decisions made by the AutoRef by comparing the rule violations to his decisions on incidents in the game by watching the recorded videos.

2. Stakeholders and Concerns

This chapter introduces the stakeholders interested in this project, their main concerns and the prioritised concerns that we would address in this project. The prioritised stakeholder concerns have been approved and verified by the stakeholders in the weekly stakeholder meetings. The identified stakeholders are analysed as shown in the figure 2.1 and concerns of the high-power and high-interest stakeholders are addressed. The bullet points are prioritised from most important to least important(good to have).

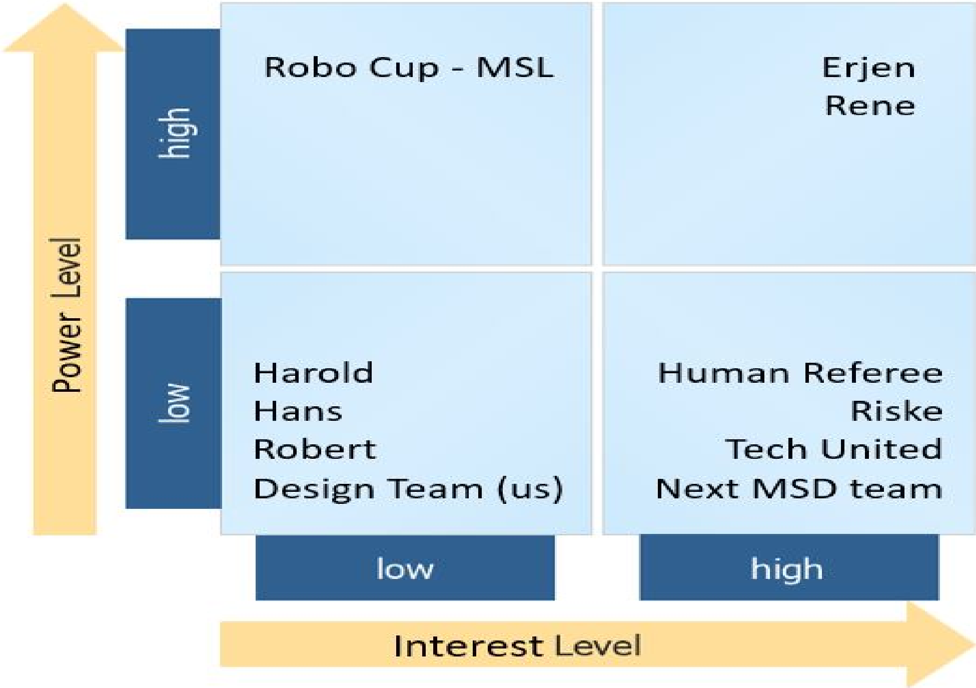

Figure 2.1: The figure shows the high-power and high-interest analysis for the identified stakeholders in the project.

2.1 Prof. dr. ir. René van de Molengraft

Power - Interest: Rene has high power and a high interest in the project.
Relationship: Rene is one of the owner of the project and has also been the owner of the project through out the years. He is also the founder of the Tech United club, who is again one of the most important stakeholder in that is interested in the project.

2.1.1 Concerns of René

1. Structured and Systematic approach to the design of the AutoRef so that the future MSD teams could continue with design activities aligning with the current approach than to start everything from scratch.
2. Extend-ability of the AutoRef in the future.
3. AutoRef should be developed and designed at the right abstraction level.
4. Integration of AutoRef with Tech United so that more people can work on it and contribute.
5. AutoRef to referee 5 mins match with 2 vs 2 turtle robots.

6. AutoRef to get positive feedback on its decisions from the human referee.
7. AutoRef should log and show the evidence for the called rule violations.

The views considering the prioritised concerns (1, 2 and 3) of René are provided in Chapter 4 Models and Views, under the sub-section 4.1.

2.2 Prof. dr. ir. Erjen Lefeber as product owner

Power - Interest: Erjen has high power and a high interest in the project.
Relationship: Erjen is one of the owner of the project and has also been the owner of the project since a last few years. Erjen is also the guide and expert on drone modelling and controls.

2.2.1 Concerns of Erjen

1. AutoRef to referee 5 mins match with 2 vs 2 turtle robots.
2. AutoRef to get positive feedback on its decisions from the human referee.
3. AutoRef should log and show the evidence for the called rule violations.
4. To reuse the previous work and Extend-ability of the AutoRef to future work.
5. Software delivery at the end of the project as a repository, with a wiki page.
6. Documentation delivery at the end of the project.
7. Individual learning goals of the trainees.

2.3 Prof. dr. ir. Erjen Lefeber as human referee

Power - Interest: The human referee has low power and a high interest in the project.
Relationship: Erjen is one of the owner of the project and has also been the owner of the project since a last few years. Erjen is also part time soccer referee, so he would also wears the hat of human referee and discusses issues that need to be addressed from a human referee's perspective.

2.3.1 Concerns of Human Referee

1. The loss of human referee's job due to the AutoRef.
2. AutoRef should help the human referee in rules that are difficult for the human referee to interpret.
3. AutoRef should log and show the evidence for the called rule violations.
4. AutoRef should be fair and ethical in refereeing.
5. AutoRef should perform on all types of robot players.
6. To have communication with the Autoref.

2.4 Other Stakeholders

The Tech United club, Riske, Robert, Harold, the Design team and the future design team are some of the stakeholders who has a high interest in the project.

3. Environment Analysis

The autonomous referee is designed to referee a robot soccer game containing 2 against 2 robots. Therefore, the application environment should remain consistent to minimise negative impacts to the interoperability and quality of the project. The boundaries of the AutoRef project are defined in accordance to the scope of the project and the technical design. The interactions between the environment and the AutoRef is presented below:

3.1 TURTLE

The Tech United RoboCup Team: Limited Edition (TURTLE) is a soccer robot player that contains the sensors responsible to generate the data for the AutoRef decision-making algorithm. The robot consists of 2 main software modules, vision and motion, and 1 hardware module containing actuators and sensors. The communication between TURTLE and Tech United repository is made via WiFi. The detailed explanation about TURTLE can be found in the document "TURTLE feasibility".

3.2 Tech United repository

Interfaces the communication between TURTLE and AutoRef. It is responsible to collect the raw data from the sensors and process it in a .mat file that will be used by AutoRef for the decision maker module.

3.3 Stationary cameras

5 surveillance cameras model Bosch Flexidome IP Starlight are used to record the match for validation of the tasks implemented. The cameras are placed on the top of the field and are used to validate a violation identified by the AutoRef. The communication between the stationary cameras and the Referee's laptop is made via WiFi. The detailed explanation about stationary cameras can be found in the document "Camera feasibility".

3.4 Tech United soccer field

Robot soccer field located at the TU/e Robotics lab. The field has the following measures: Length 12 m and Width 8 m. The penalty area has 1.5 m length and 3 m width.

3.5 Human referee

Receives the information from AutoRef about rule violation and gives positive or negative feedback to the decision-making algorithm. The Human referee communicates to the TURTLE via Refbox.

3.6 Refbox

The Referee Box (refbox) controls, monitors, and evaluates the game. The referee box also logs each and every message sent or received, all internal state changes of the knowledge-based system, as well as game-relevant reports and information. In future implementation, Refbox can be able to communicate with AutoRef.

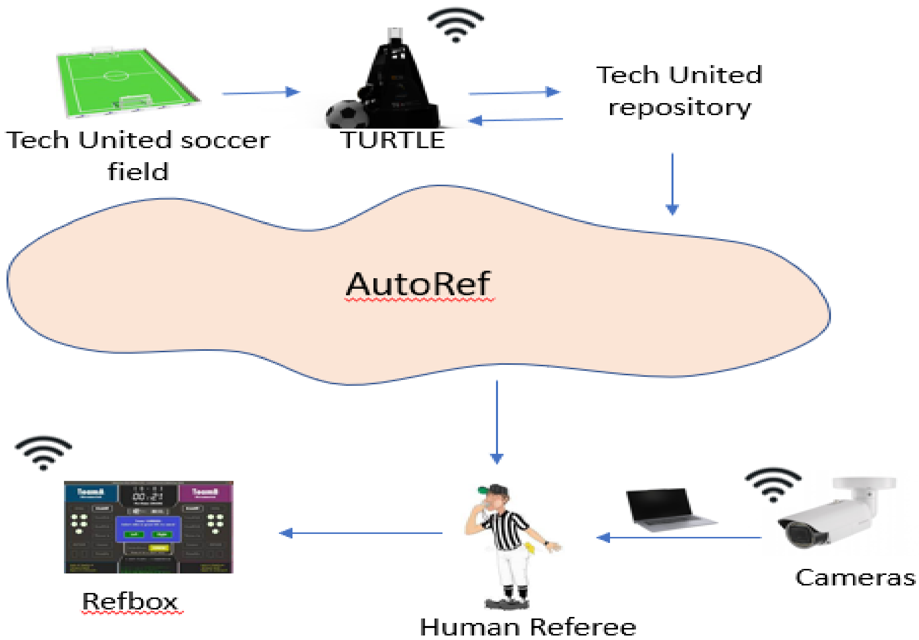

Figure 3.1: The figure shows the systems that interact with AutoRef creating its environment.

3.7 Future Work and Maintenance

The work that has been done in this project is to give a proper and usable documentation, more autonomy, more dimensions to extend this project to the future teams working on it. The below subsections give a brief introduction on how this project could be extended in the future.

3.7.1 Perception

Data

The data is generated by the Tech United code. The scripts take the data from the turtles(soccer players) for the duration of the match and use this data to make decisions, assumptions are made that there is no bias in the data.

A good place to start is to think of having an independent data collection strategy which has no bias with respect to the teams. A few data source options are listed below:

1. Static Cameras: Cameras could be very rich in the data, that could be used for the AutoRef. A feasibility study has been carried out to investigate the usage of the cameras. The camera feasibility report [9] document would be a good start.

2. Drones: Drones are complicated systems, CrazyFlie 2.1 could be used for AutoRef. The drones bring in a lot of challenges regarding controls, path planning, battery, available camera hardware for drones and finally controlling one or more drones to capture the entire field and estimate the whole ball. A feasibility report is also carried out on drones here [8].

3. Turtles: Turtles are one of the most capable hardware that is easily available, there is also a lot of support that is available for turtles, [2] is a great place to ask for support, also they have a significant interest in this. This project just collected the data from players(turtles) to make decisions, one or more turtles could be used as the referee, that could see players and

balls. This will also solve the problem of bias in the data, as you now have an independent collection system for the soccer game. **Continuing, with turtles is also one of the primary concerns for the stakeholders(project owners).** The feasibility analysis on turtles is found here [4].

4. Fusion of data from Multiple Sources: One more approach is to fuse the data from multiple sources to make the decisions as an AutoRef. This can help you to capture the game from multiple perspectives. The mathematical complexity involved in this is a bit overwhelming for the duration of the project. This could be termed as a research project than a design project.

Processing

Perception consists of gathering data and then processing data to extract meaningful information. The above sections gives an introduction to the data sources that could be used to gather data from the game. This section explains the challenges in processing the data and potential starting points to improve the data processing to improve the Autoref. The data that is being used is complicated in nature, it is hard to understand the data that is associated with the players and the ball etc because of lack of documentation from Tech United. The whole process of data collection from the players could be optimised with novel data collection techniques so that the further processing could be easy and less time consuming.

The estimation algorithms for player positions, ball positions, velocities, ball tracking etc. could be improved by using more reliable and advanced filtering or smoothing techniques. Different deep learning techniques could also be used for ball and player detection.

3.7.2 Cognition

Decision Maker

The designed AutoRef has three tasks or three capabilities, 1. To detect if the ball is in the field or out of the field, 2. To detect if the ball has rolled for at least a minimum of 0.5 m at different game states. 3. Assign possession to the team and player, when the possession of the ball is lost by kicking the ball outside the pitch.

Rules

Different rules could be analysed and then the capabilities of the AutoRef could be improved or added to detect more rules violations for the robot soccer game.

On-line or Off-line

The designed AutoRef works offline, makes the decision offline. This could be where the future teams could work on. The AutoRef could be developed to make the decisions in Real-Time, when the robot soccer match is being played live.

Algorithm

The whole process of making decisions could also be improved on the current algorithms, adding confidence levels, indicating the confidence in making the rule violation decision. An alternative could be to use Reinforcement learning or Multi-Agent reinforcement learning to learn how to make the decisions by experience. The developed AutoRef is developed in Matlab, this implementation could also be revisited to investigate if implementing this in different programming languages makes sense. For instance, if this could be developed in C, C++ or python it could be deployed on hardware.

3.7.3 Communication

Communication with RefBox

RefBox is a communication interface between the human referee and the robots. In this project the communication with RefBox or with the human referee was avoided because of the simplicity and the available time. The AutoRef just indicates the rule violations and logs the data of the game state, as to when the violation occurred. Having a communication with the RefBox or integrating the AutoRef with RefBox will be a good start.

Real-Time communication with Human Referee

The AutoRef should be capable of helping the human referee or independently carryout refereeing for a robot soccer match. Real-Time communication with a Human Referee would also be a pre-requisite to have for the AutoRef.

4. Models and Views

This chapter explains how the stakeholders' concerns that are defined in chapter 2 are addressed from different views.

4.1 Concerns of René

4.1.1 Concern 1

The first concern of René is "Structured and Systematic approach to the design of the AutoRef so that the future MSD teams could continue with design activities aligning with the current approach than to start everything from scratch. "

View: Systematic Approach

This could be addressed by explaining our rationale behind key design choices. This could perhaps motivate and enable the future teams to understand the design choices based on the rationale provided, align with the approach detailed by MSD 2021, and in turn improve the design effectively, and efficiently than previous years. The key decisions and the rationales are listed below:

- Usage of Turtles: We got incredible support on turtles. The most of the design or capabilities or development of turtles is already done. We just found what all could be reused. We conducted feasibility studies on all available options and evaluated each of these options in a Pugh matrix with defined criteria that was derived from the stakeholder concerns.
- Scope: The scope was defined such a way that, the future teams can easily contribute to the AutoRef. AutoRef can be developed further in various dimensions.
- Platform: Matlab was used because of several reasons, team skill competency in Matlab, Most of the Tech United code was in Matlab, Past teams worked in Matlab.
- Modular Design: We chose to design the Autoref as a Modular Design to give maximum autonomy to reuse and extend the AutoRef.
- Development and Integration: Functions were developed from functional decomposition and DSM was used to identify the relationships. This helped us to develop the functions efficiently with available time. The code was developed robustly such that the algorithms work on data that is collected from the game and is independent of the data source. The code was integrated such a way that is easy to extend the AutoRef in different dimensions, particularly with adding more tasks to the existing system. The future sections explain this in detail.
- Relationship with Tech United: The interaction with Tech United is highly appreciated by the stakeholders. Tech United also has very talented members who are willing to help and could also provide valuable insights into the problem.

4.1.2 Concern 2

The second concern of René is "Extend-ability of the AutoRef in the future". This concern has been the most important part of our development and is addressed from multiple views as given below:

View: Extend-Ability of main functionalities

The figure **??** shows the different blocks of the AutoRef, and the section 3.7 explains how contributions could be made in the future building on the existing work.

View: Extension - Adding More Rules

This section explains how to add more rules to the existing AutoRef. This problem has been given enough thought, so that adding more rules should be less time consuming, easy and not making major changes to the code structure. The below guide gives a step by step procedure to add more rules to the AutoRef.

1. Identify the task that needs to be performed for monitoring the new rule or rules.
2. Check if a task is already implemented. If not, specify it in terms of skills and conditions under which the task must be performed.
3. Identify the "skills" that are needed for the task.
4. Check if the skill or skills are already present. If not, specify what the skill is and check if you can use existing skill to implement them.
5. Identify if the skill is explicitly related to the task or the skill is explicitly related to the data. If it is more related to the data then add the skill to the DataProcessing class else keep the skill local to the Task function.
6. Implement each new task as a separate function.
7. Test the tasks.
8. Define Test scenarios or create such scenarios in a robot soccer match.
9. Validate the tasks with Robot Soccer matches.
10. Integrate the tasks to the existing AutoRef.
11. Validate the updated AutoRef.
12. Document the changes made to the AutoRef.

View: Extension - Interfacing AutoRef with RefBox

The existing AutoRef could also be interfaced with RefBox, so as to communicate the AutoRef decisions to the Human Referee or to read the game state(kick-off, half-time, throw-in, corner etc.) information from the RefBox. The below guide gives a step by step procedure to interface the AutoRef with the RefBox.

1. Identify the key need: a) communicate AutoRef to human referee, b) read game state data from the RefBox.
2. Develop this skill as a function in Matlab.
3. Unit test this function to ensure it is working.
4. Once this function is working as intended, make this function as the "method" of the DataProcessing class.
5. Call this functions when neccessary in the Task functions.
6. Document the changes made to the AutoRef.

View: Extension - Adding More Robot Players

The design and the developed code is robust to the number of robot players in the team. If in the future, more robot players are added to the teams, the code still functions the way it intended to function for 2 vs 2 players.

Adding more robot players needs no changes in the code, but you should add more player robots in the lab when the match is being played.

View: Extension - Re-usability

The below figure shows the code implementation for the AutoRef. The Data Processing is implemented as a class file, while the Task 1 and Task 2 are implemented as functions. If the future teams decide to collect data from other sources, the Data Processing class could be inherited or extended to process that data and the task functions will still work fine.

The future teams could also work on adding more tasks to the AutoRef, for this they could develop more functions and integrate it, this will improve and enhance the AutoRef in terms of skills and autonomy.

The UML diagram below 4.1 shows the code integration and implementation diagram for the AutoRef. There are two types of boxes, viz. 1. Dataprocessing 2. Task*. The DataProcessing box is a class file it shows the class the data fields of the class and the properties of the class. The "#" before the file name shows that the fileName is public. The other type of boxes for Task1 and Task2 are the functions that implements the skill of the AutoRef. In future more functions could be developed to add more skills to the AutoRef.

The relationship type between the class and the functions is of type aggregation, as shown by the arrow, it means that there is an association between class and the functions, but the class can exsist without the Task* functions. And the same class could be used to implement many such Tasks*, but the Tasks* cannot exsist without the class. + are the functions of the class, - are the functions developed locally to fulfill the respective individual tasks.

Tasks* = Task1(), Task2()

Note: Additional explanation on the decomposition of Task 1 and Task 2, and the code development strategy is explained in A.3.

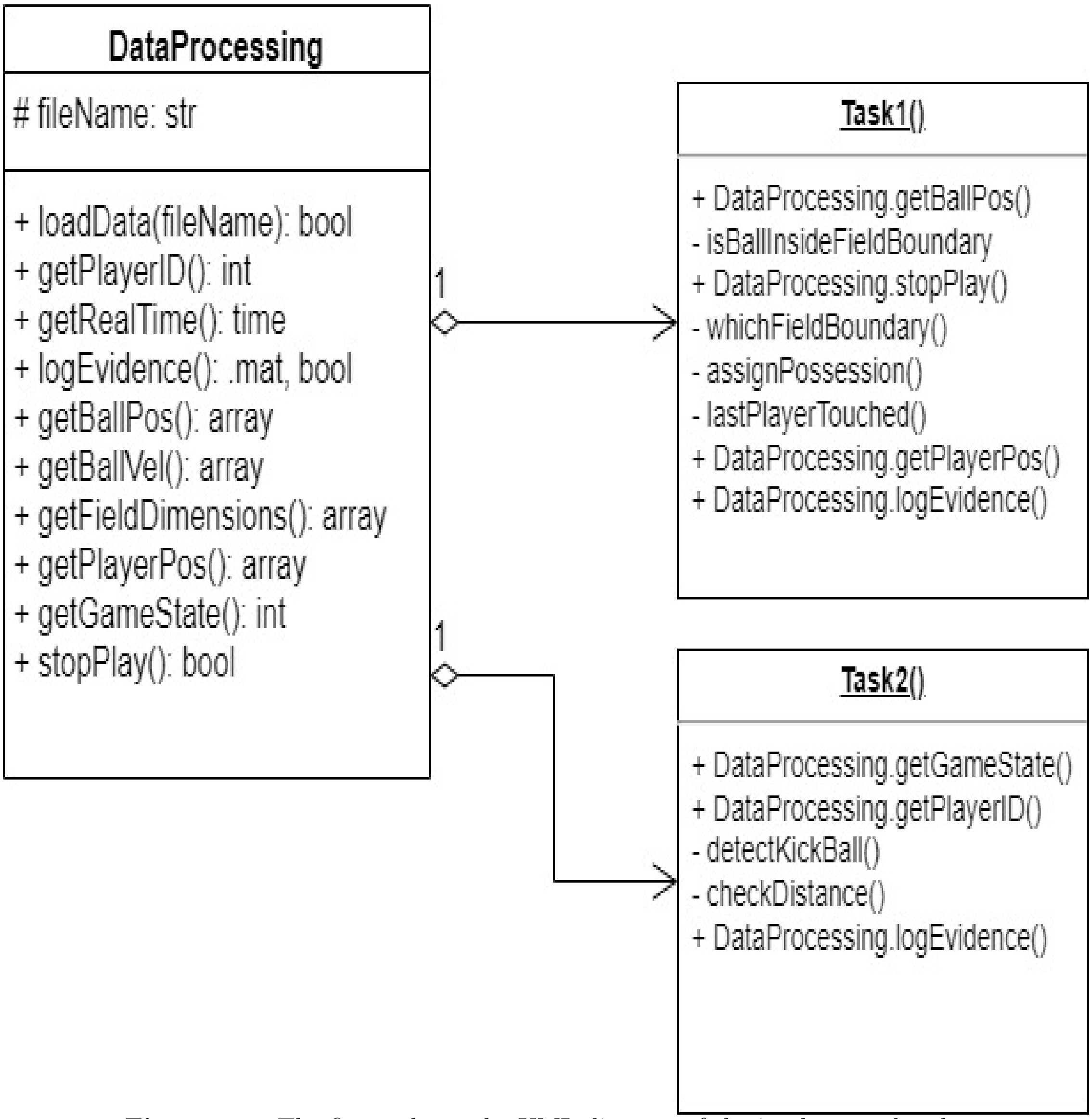

Figure 4.1: The figure shows the UML diagram of the implemented code.

4.1.3 Concern 3

The third concern of Rene is “AutoRef should be developed and designed at the right abstraction level.”

View:Right Abstraction Level

The below figure 4.2 shows the decomposition of the AutoRef as components. The figure is read as, AutoRef should carry out 1 or more tasks, one task could lead to one or many rule violations. The level below shows the necessary dependencies that is needed to perform a task, that is we need at least data from 1 game, which is made up of 2 teams, 1 field, 1 ball and 1 team could have 2 players.

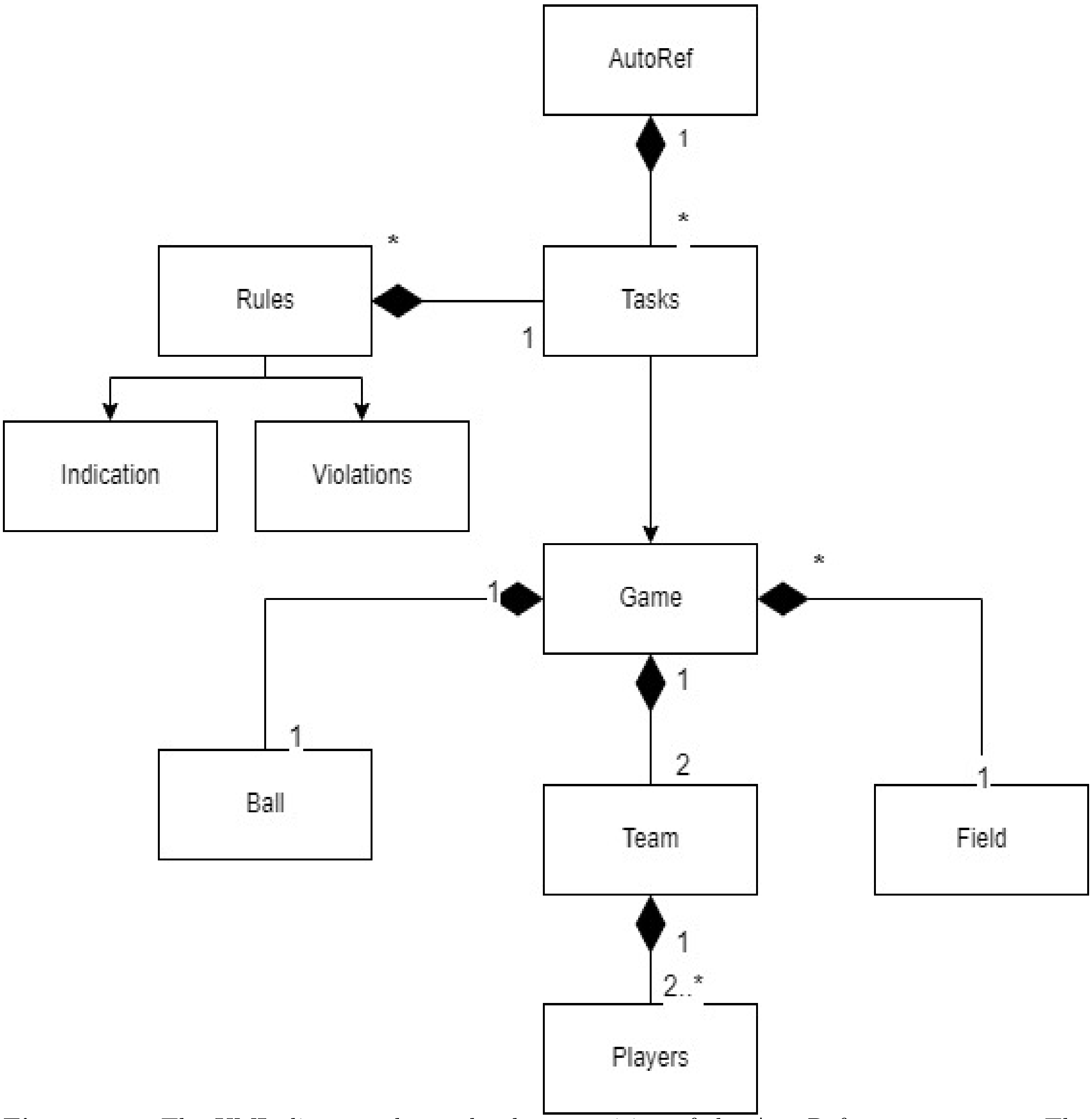

Figure 4.2: The UML diagram shows the decomposition of the AutoRef as components. The project designed at a very high level, meaning that AutoRef is made up to Tasks and we just focus on designing the tasks and deliver a common coding platform to add more tasks in the future. The block below task, Game shows the data needed for the tasks, different data from different sources could also be used for tasks in the future.

Bibliography

[1] Erjen Lefeber Rene van de Molengraft. AutoRef — Autonomous Referee System. 2022. 1

[2] Tech United. Tech United. 2022. 1, 11

[3] Robocup Middle Size League. Rule and Regulations. 2021. 1, 4

[4] Kian Marzhan. TechUnited Turtle Feasibility Analysis. 2022. 2, 4, 12

[5] Gerard Brand. Referee myth-busting: How many decisions do officials get right? 2022. 2

[6] wikipedia. Assistant referee (association football). 2022. 3

[7] Meghna Yuri. TechUnited Turtle Feasibility Analysis. 2022. 4

[8] Shubo Cevahir. Drone Possibilities. 2022. 4, 11

[9] Shubo Cevahir. Feasibility Report of Stationary Camera. 2022. 4, 11

[10] Meghna Mohamed. AutoRef System Requirements. 2022. 4

[11] Meghna Mohamed. Task Requirements. 2022. 5

A. Appendix

A.1 Glossary

This section provides a brief description of the terms that are used in the context of AutoRef project.

1. **AutoRef** - AutoRef stands for Autonomous Referee which would assist the human referee in refereeing RoboCup games.

2. **Rule book** - Middle Size Robot League Rules and Regulations for 2022, Version - 23.0 20211231. Rule book consists of the FIFA laws and corresponding RoboCup modifications, Competition rules, and F - 2000 challenge.

3. **Law/Rule** - Laws are the FIFA laws and/or corresponding modifications wrt RoboCup. The term “RULE” is also sometimes interchangeably used.

4. **Violation** - Violation corresponds to “not adhering the law/rule” as provided in the rule book and identified by human referee or AutoRef.

5. **Task/Function/ Capability** - Task is the objective of AutoRef to identify one or more sets of rule/law violations. The term “Function/(s) or “Capability/(ies)” are also sometimes interchangeably used.

6. **Skill** - Skills are a set of sub - functions required to be performed by the AutoRef in order to accomplish a Task. The skills need not be exclusive to one particular task. The term “is also sometimes interchangeably used.

7. **Dimension** - The term dimension used in Chapter 1.4 Technical Design point 3 under the sub - section Considerations, denotes the extend-ability of the AutoRef project in two directions. One could be adding and implementing additional rules where as the other could be improving the efficiency of the current algorithm and software by including factors like confidence bound, error margin, etc.

8. **Scenario/(s)** - The term scenario/(s) used in Chapter 1.4 Technical Design point 1 under the sub - section Considerations, represents the game situations that were created or used in order to verify the correctness of the algorithm that would identify rule violations. It could either be by hard coding the position coordinates for the ball and players as done by MSD 2020 or by making use of real game data from the player robots as done by MSD 2021.

9. **Data** - Data in the context of AutoRef project refers to the information collected from different sources with respect to the RoboCup game, which would be used in decision making algorithms, to determine whether or not laws/rules are violated. For the MSD 2021 AutoRef, for the implementation, data corresponds to the .mat file consisting of information like player position, ball position, ball velocity, refbox command, field dimensions, time information, etc., which is collected from the RoboCup players - TURTLES.

10. **Off-line** - The term Off-line mentioned in Chapter 1.5 Realisation under the sub-section Cognition is in the context that the implementation by MSD 2021 for decision making, to determine whether or not laws/rules are violated, is done using the data collected from the game in the format of .mat file once the game is completed. The decision making for rule violation is not currently done with data streaming real time while the game is ongoing.

11. **On-line** - The term On-line mentioned in Chapter 1.5 Realisation under the sub-section Cognition is in the context that the decision making for rule violation could be improved by making use of data which could be streamed real time while the game is ongoing. This is a suggestion for improvement on the existing AutoRef delivered by MSD 2021.

A.2 Open questions for future teams

1. The right abstract level should be defined with stakeholders. The team addressed it as a concern but did not define until which level the project should be developed.

2. The extend-ability of the project should be defined with stakeholders by defining what should be extendable in the project.

3. All stakeholders concerns should be addressed during the project development. The 2021 MSD team addressed the 3 main concerns of René. The next generation should define which concerns will be considered in the development.

4. The structure and the type of systematic approach should be defined with the stakeholders.

5. The integration of the AutoRef and Tech United software should be further discussed with stakeholders.

A.3 Functional Decomposition and DSM

A.3.1 Task 1

The below A.1 figure shows the functional decomposition and the development plan for the laws 9.1, 9.2, 15* of the MSL Rulebook 2022 V23.0

The big square blocks define the functions that will be implemnted in Matlab, the bold characters show the function name, the arguments needed for the function and the logic of the function is developer's freedom, the text in blue shows the variables returned by the function and the font in orange are the short synonyms for the function. Ex. The output of the functions T1F1 and T1F3 are used for the function T1F4.

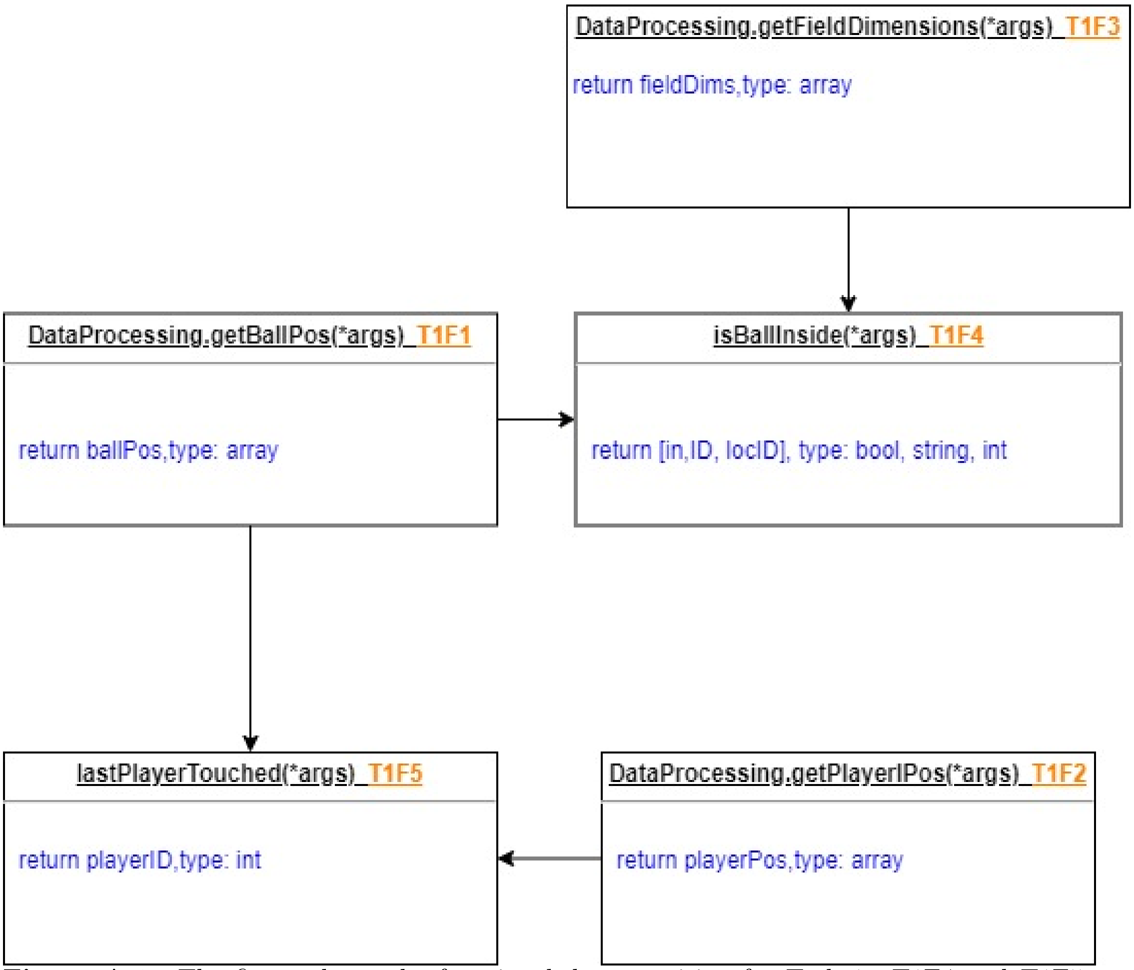

Figure A.1: The figure shows the functional decomposition for Task 1. T1F4 and T1F5 are functions(skills) that are local to the task.

A.3.2 Task 2

The below figure A.2 shows the functional decomposition and the development plan for the laws 8.3, 15.1, 16.1,17.1 of the MSL Rulebook 2022 V23.0

The big square blocks define the functions that will be implemented in matlab, the bold characters show the function name, the argumnents needed for the function and the logic of the function is developer's freedom, the text in blue shows the variables returned by the function and the font in orange are the short synonyms for the function names. The arrows define the information dependency relationship between the functions. Ex: The function f4 calls f1, f2 and f3, and uses the returned information to feed f5.

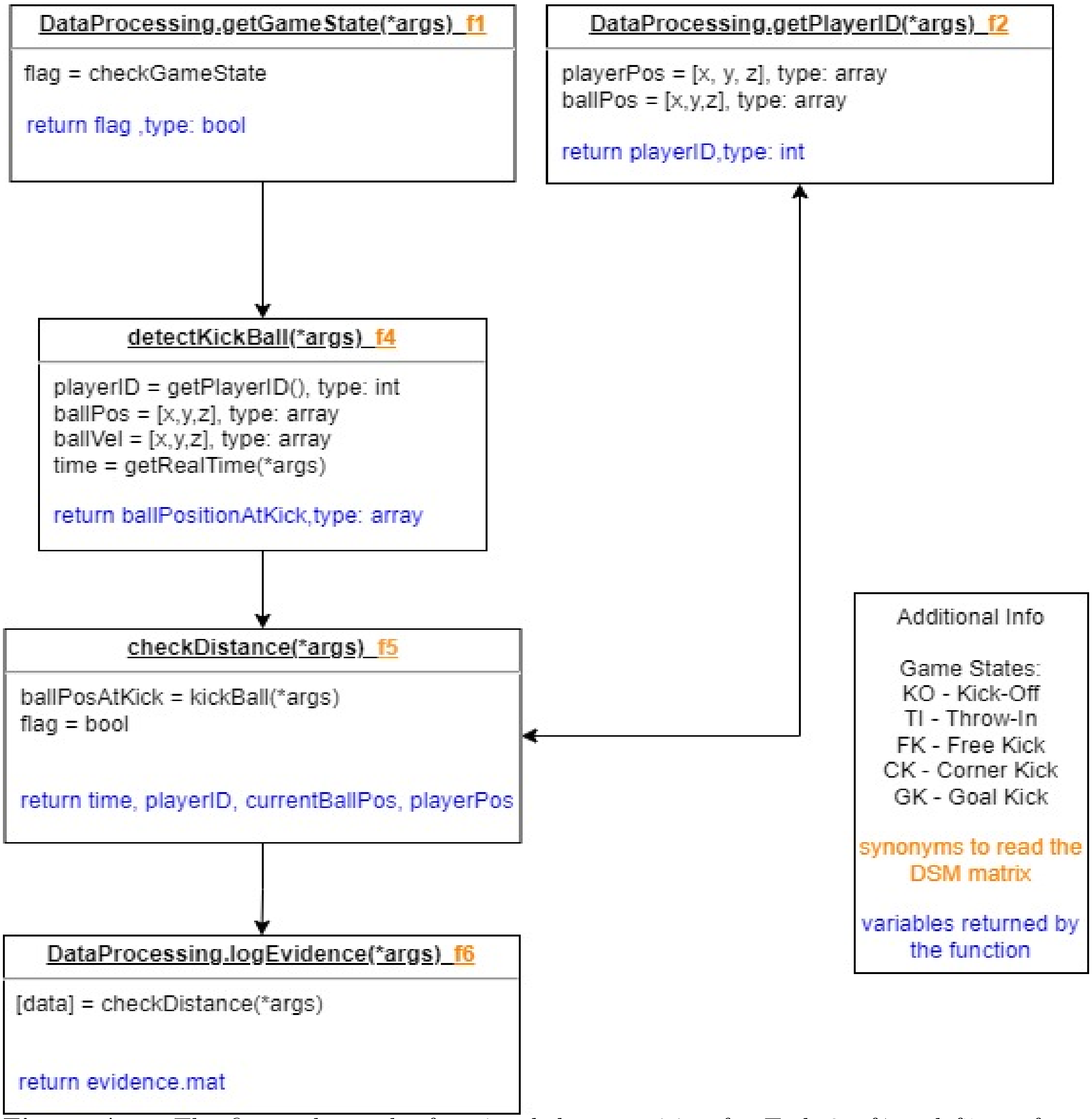

Figure A.2: The figure shows the functional decomposition for Task 2. f4 and f5 are functions(skills) that are local to the task.

Development Process Optimisation

DSM were used to analyse the relationships between the functions to make an optimal code development plan. The below figure A.3 shows the DSM that is drawn for the Task 2. There is a high correlation between functions f4 and f5.

Note: The function f3 was called getRealTime(), we found later that the task 2 could also be solved without the notion of real time. Hence the updated functional decomposition shown in A.2 does not have the function f3.

Note: The figure uses ICFBD(**I**nput in **R**ows, **F**eedback **B**elow the **D**iagonal) notation. Meaning that the function f4 feeds its output to f5 and f5 takes input from f4, f3 and f2.

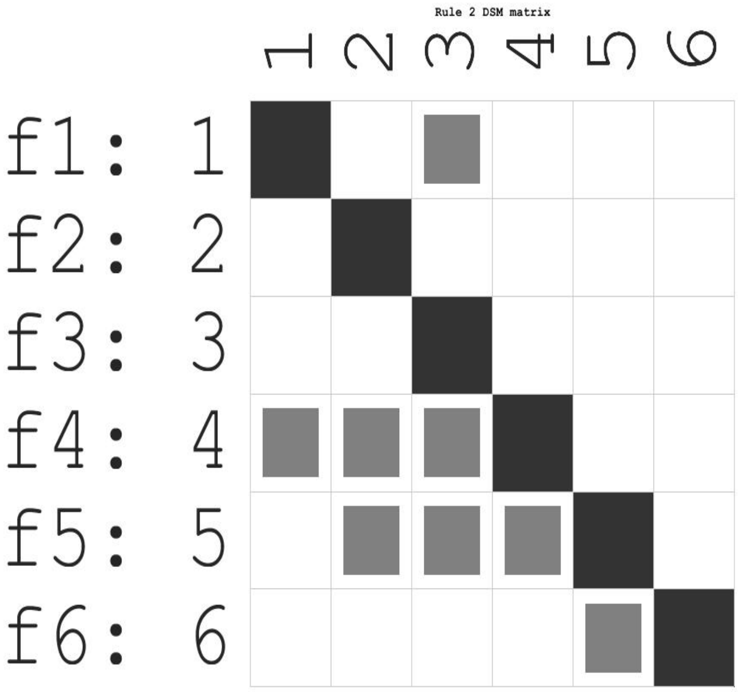

Figure A.3: The DSM for Task 2. The variables(f1,f2,..f6) correspond to the synonyms used in the functional decomposition diagram shown in A.2.

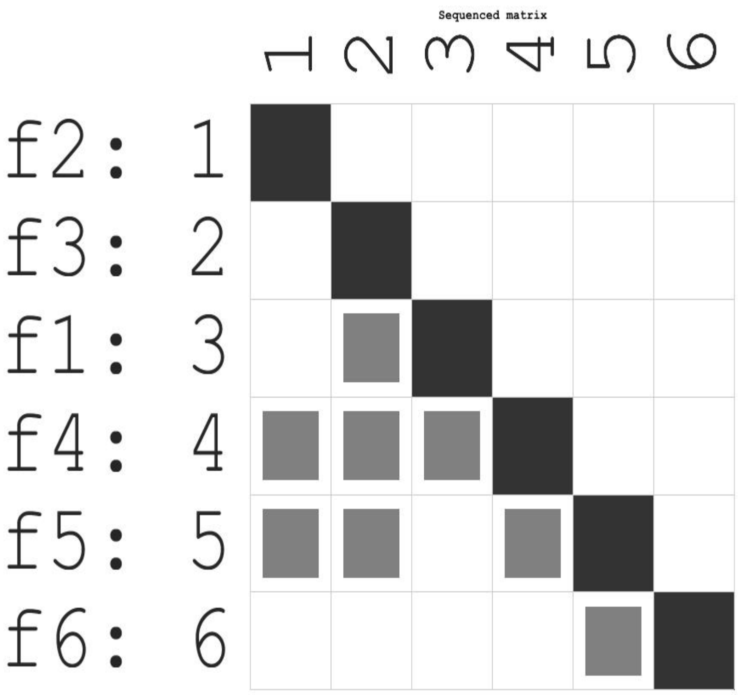

Figure A.4: Optimised and Sequenced DSM of figure that is in the above figure A.3

The above figure A.4 shows the sequenced and optimised sequence. We can see that f1, f4 and f5 are sequenced together as there is some information exchange between the functions. Therefore, f1, f2,f3 and f6 are developed in parallel, while a team of 3 members worked on sequentially developing functions f4 and later f5.