

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การศึกษาและพัฒนา Agent สำหรับการแข่งขัน Robo Cup
Study and develop agent team for Robo Cup

โดย

นายสุรเชษฐ เดชระฆังทอง

รหัส 42067166

วัน เดือน ปี.....	19 มิ.ย. 2550
เลขทะเบียน.....	01866
เลขเรียกหนังสือ	วทท. ส. 848ก 2544
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."	

อาจารย์ที่ปรึกษา

ศศ. ดร. โชติพงษ์ ภรณ์ฉลวย



H001866

รายงานนี้เป็นส่วนหนึ่งของโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2544

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การศึกษาและพัฒนา Agent สำหรับการแข่งขัน Robo Cup
นักศึกษา	นายสุรเชษฐ เคนระฆังทอง
อาจารย์ที่ปรึกษา	ผศ. ดร. โชติพัทธ์ ภรณ์วลัย
ระดับการศึกษา	วิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2544

บทคัดย่อ

ฟุตบอลจัดเป็นเกมสกีการแข่งขันทที่มีความสลับซับซ้อน และเป็นกีฬาประเภททีมที่สร้างความสนใจให้ผูชมได้มากที่สุดเท่าได้ จึงได้มีการจัดการแข่งขันฟุตบอลที่จำลองลงบน Software Computer ที่เรียกว่า Soccer Server ภายใต้การแข่งขัน Robo Cup โดยโครงการนี้จัดทำขึ้นเพื่อศึกษาและพัฒนาการวิเคราะห์และออกแบบ Agent สำหรับใช้เป็นผู้เล่นในการแข่งขัน โดยมีการศึกษาโปรแกรมของทีม CMUnited99 และ FCPortugal ในส่วนของ Low level Source Code แล้วนำมาเป็นแนวทางในการวิเคราะห์และออกแบบ Agent ใหม่ เพื่อให้การพัฒนาเป็นไปอย่างสะดวก และรวดเร็ว

Title	Study and develop agent team for Robo Cup
Student	Mr. Surachet Detrakangtong
Advisor	Asst. Prof. Dr. Chotipat Pornavalai
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2001

ABSTRACT

Soccer is a very complex game and the most popular team sport in the world. Soccer server is the software computer for simulation soccer under the RoboCup competition. This system development project presents the analysis and design of the agent team. The first step is to study the existing the low level source code and analyze for CMUnited99 and FCPortugal team. Then, design a new agent that can work faster, more easily and more accurately.

กิตติกรรมประกาศ

ขอขอบคุณ

บิดา มารดา และน้อง ที่ให้กำลังใจในการศึกษาตลอดเวลา

ผศ. ดร. โชติพัชร ภรณ์วลัย ผู้ให้คำปรึกษาและข้อเสนอแนะต่าง ๆ สำหรับการทำงานโครงการพัฒนา Agent
ที่ใช้ในการแข่งขัน RoboCup

เพื่อน ๆ ที่ให้กำลังใจและคำปรึกษาตลอดระยะเวลาที่ศึกษา

สุรเชษฐ เศรษฐะมั่งทอง

มีนาคม 2545



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	
บทที่	
1. บทนำ.....	1
1.1 วัตถุประสงค์.....	1
1.2 ขอบเขตของการพัฒนาระบบ.....	1
1.3 หลักวิธีที่ใช้ในการพัฒนาระบบ.....	1
1.4 เครื่องมือที่ใช้ในการพัฒนาระบบ.....	1
2. ทฤษฎีที่เกี่ยวข้อง.....	2
2.1 ทฤษฎีพื้นฐานของ Multi Agent System (MAS).....	2
2.1.1 Homogeneous, Non-Communicating MAS.....	2
2.1.2 Heterogeneous, Non-Communicating MAS.....	3
2.1.3 Homogeneous, Communicating MAS.....	3
2.1.4 Heterogeneous, Communicating MAS.....	4
2.2 กลไกในการติดต่อสื่อสาร.....	4
2.2.1 Stream communication.....	5
2.2.2 Datagram communication.....	6
2.3 หลักการและทฤษฎีที่นำมาใช้ในทีม FC Portugal.....	6
2.4 การวางแผนของทีม (Team Strategy).....	7
2.4.1 SBSP - Situation Based Strategic Positioning.....	7
2.4.2 ชนิดของผู้เล่น (Player Type).....	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
2.4.3 DPRE - Dynamic Positioning and Role Exchange.....	8
2.5 Agent Architecture and High Level Decision Module.....	9
2.5.1 World State.....	9
2.5.2 Action.....	10
2.5.2.1 Basic action.....	10
2.5.2.2 Advance action.....	11
2.6 Intelligent Perception and Communication.....	11
2.6.1 ADVCOM Intelligent Communication Mechanism.....	12
2.6.2 SLM Strategic Looking Mechanism.....	12
2.7 Integrating Soccer Knowledge.....	12
2.7.1 Ball Possession.....	12
2.7.2 Ball Recovery.....	13
2.7.3 Goalie Defensive Strategy.....	13
2.7.4 Low-Level Skills – Optimization Kick.....	14
3. การทำงานของ Soccer Server	15
3.1 Soccer Server.....	15
3.2 Soccer Monitor.....	15
3.3 การเรียกใช้งาน Soccer Server.....	15
3.4 Sensor Information.....	19
4. การพัฒนา Agent Team.....	20
4.1 Agent Architecture and Flow Control of the High Level Decision Module...	20
4.2 Process การทำงานของโปรแกรม FCPortugal Agent.....	22
4.3 ส่วนประกอบของโปรแกรม.....	24
4.3.1 รายละเอียดของ file ต่าง ๆ ที่ใช้ภายในโปรแกรม.....	24
4.4 การพัฒนา KMITL Agent Team.....	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
4.5 การ Compile Client Program.....	26
4.6 คำสั่งในการพัฒนาโปรแกรม Agent.....	26
5 บทสรุป.....	28
5.1 ผลการพัฒนา Agent	28
5.2 ประโยชน์ที่ได้รับ.....	29
5.3 ข้อเสนอแนะ.....	29
บรรณานุกรม.....	30
ประวัติผู้เขียน.....	31



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1	โครงสร้างของ Homogeneous, Non-Communicating MAS..... 2
2.2	โครงสร้างของ Heterogeneous, Non-Communicating MAS..... 3
2.3	โครงสร้างของ Homogeneous, Communicating MAS..... 3
2.4	โครงสร้างของ Heterogeneous, CommunicatingMAS..... 4
2.5	การติดต่อสื่อสารกันระหว่าง Client กับ Server ผ่านทาง Socket..... 5
2.6	Stream communication..... 5
2.7	Datagram communication..... 6
2.8	Objects type hierarchy..... 9
3.1	การตั้งค่าสั่ง configure..... 16
3.2	การตั้งค่าสั่ง sserver..... 16
3.3	สนามที่สร้างจาก Soccer Monitor..... 17
3.4	การใช้คำสั่งให้ CMUnited99 ลงสนามด้วยคำสั่ง Start..... 17
3.5	ภาพหลังจากการใช้คำสั่งลงสนามของทั้ง 2 ทีม..... 18
3.6	ภาพหลังจากการใช้คำสั่งลงสนามของทั้ง 2 ทีม..... 18
4.1	ภาพ Agent Architecture..... 20
4.2	ภาพ Control Flow of the High Level Decision Module..... 21
4.3	การทำงานของ Program FCPortugal Agent..... 22
4.4	การตัดสินใจเมื่อมีบอลของ Team KMITL Agent..... 23
4.5	การจัด Position ของ Team KMITL Agent..... 23
4.6	Process การตัดสินใจของ Goal Team KMITL Agent 24
4.7	ภาพการแข่งขันระหว่าง KMITL กับ CMUnited 25
5.1	การแข่งขันระหว่าง ทีม KMITL กับทีม FCPortugal..... 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 วัตถุประสงค์

- (1) ทำการศึกษาโครงสร้างการทำงานพื้นฐาน สำหรับการทำงานของ Soccer Server
- (2) ศึกษาขั้นตอนการพัฒนาโปรแกรม ที่นำมาพัฒนาเป็น Agent สำหรับเชื่อมต่อเข้ากับ Soccer Server เพื่อใช้ในการจำลองการแข่งขันฟุตบอลของ Robo Cup
- (3) เพื่อเป็นพื้นฐานที่สำคัญในการนำไปพัฒนาโปรแกรม Agent ที่สามารถเรียนรู้ได้ต่อไปในอนาคต
- (4) ทำการพัฒนาโปรแกรม Agent จากโปรแกรมพื้นฐานของทีม CMUnited99 และทีม FCPortugal

1.2 ขอบเขตของการพัฒนาโครงการ

พัฒนาโปรแกรมในส่วนของ Agent หรือ Client ที่จะนำมาใช้ในการเชื่อมต่อเข้ากับ Soccer Server จากโปรแกรมและฟังก์ชันพื้นฐานที่ได้ทำการศึกษาจาก Agent ของทีม CMUnited99 และ Agent ของทีม FCPortugal

1.3 หลักวิธีที่ใช้ในการพัฒนาระบบ

ขั้นตอนการดำเนินงานในการวิเคราะห์และออกแบบ Agent โดยแนวทางของ Multi Agent System (MAS) สำหรับภาษาที่ใช้ออกแบบ คือ ภาษา C++ ซึ่งได้ใช้หลักการในการโปรแกรมด้วยวิธีเชิงวัตถุ และได้ทำการพัฒนาภายใต้ระบบปฏิบัติการ Linux

1.4 เครื่องมือที่ใช้ในการพัฒนาระบบ

ในการศึกษาวิเคราะห์และพัฒนาโครงการนี้ มีเครื่องมือที่ใช้ คือ

- (1) ระบบปฏิบัติการ Linux RedHat version 6.2
- (2) โปรแกรม Soccer Server version 7.08
- (3) เครื่องมือที่นำมาใช้ในการพัฒนาโปรแกรม คือ ภาษา C++

บทที่ 2

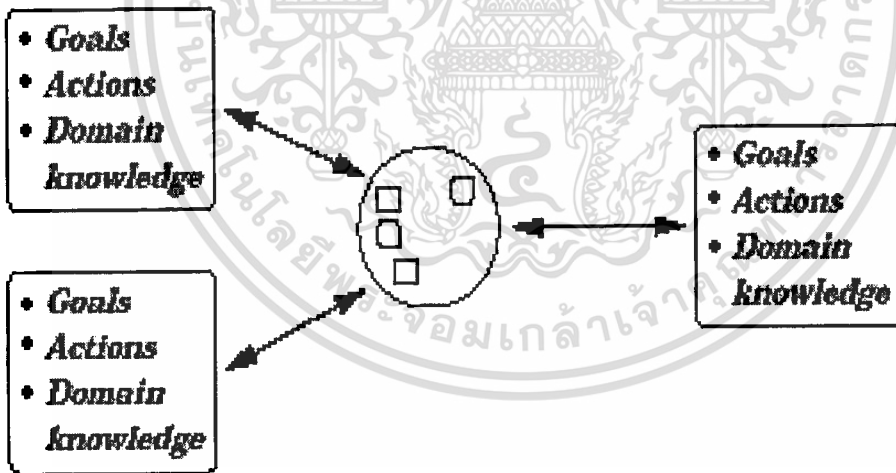
ทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีพื้นฐานของ Multi Agent System (MAS)

Multi Agent System (MAS) จัดเป็นแขนงวิชาหนึ่งของเรื่องปัญญาประดิษฐ์ หรือ Artificial Intelligence (AI) ได้มีการนำมาใช้เป็นส่วนหนึ่งของแนวทางในการพัฒนา Robotic Soccer และที่ได้นำมาศึกษาจะได้จัดแบ่งออกเป็น 4 ประเภท

2.1.1 Homogeneous, Non-Communicating MAS

Agent ทุกตัวจะมีลักษณะภายในที่เหมือนกันทุกตัวไม่ว่าจะเป็น Goals, Actions, Domain knowledge จากรูปจะสังเกตเห็นได้ว่าตัวอักษรที่อยู่ในกรอบสี่เหลี่ยมจะเหมือนกัน (แทนตัว Agent) และจะไม่มีการติดต่อสื่อสารกันระหว่าง Agent ด้วยกันเองแต่จะเป็นการติดต่อสื่อสารกับ Environment (แทนด้วยรูปวงกลม)



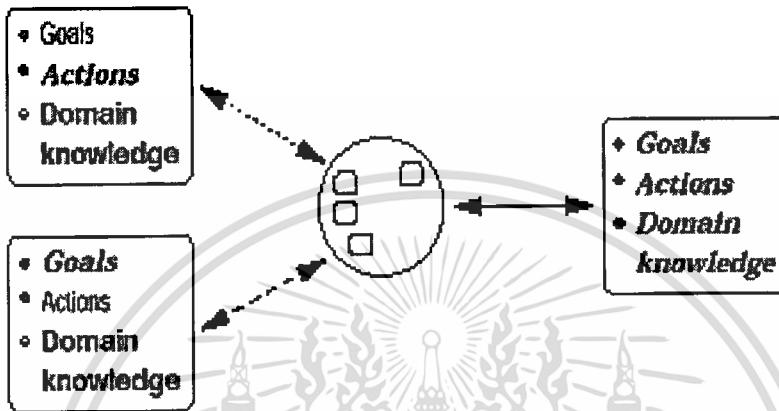
รูปที่ 2.1 โครงสร้างของ Homogeneous, Non-Communicating MAS

ที่มา Layer Learning in Multi-Agent System

2.1.2 Heterogeneous, Non-Communicating MAS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Agent ในแบบนี้จะมีลักษณะภายในที่ต่างกัน ทั้ง Goals , Actions , Domain knowledge จากรูปจะแสดงให้เห็นว่าภายในรูปสี่เหลี่ยมแต่ละรูปตัวอักษรจะแตกต่างกัน แสดงให้เห็นถึงความแตกต่างกันของ Agent แต่ยังไม่มีการติดต่อสื่อสารกันระหว่าง agent ด้วยกันเอง แต่ยังคงมีการติดต่อสื่อสารกันระหว่าง Agent และ Environment

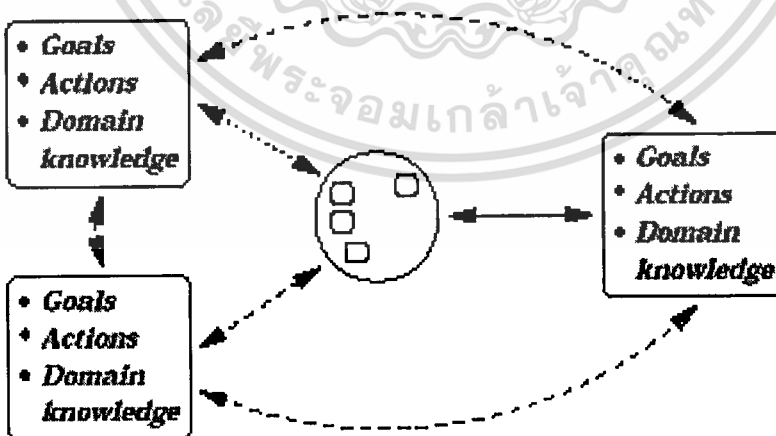


รูปที่ 2.2 โครงสร้างของ Heterogeneous, Non-Communicating MAS

ที่มา Layer Learning in Multi-Agent System

2.1.3 Homogeneous, Communicating MAS

Agent ในรูปแบบนี้จะมีการติดต่อสื่อสารกันระหว่าง Agent ด้วยกันเอง แต่ว่าลักษณะภายในของตัว Agent นั้นจะเหมือนกัน จะสังเกตได้จากตัวอักษรที่อยู่ภายใน Agent ที่แสดงให้เห็นว่ามีองค์ประกอบภายในที่เหมือนกัน และยังสามารที่จะติดต่อสื่อสารระหว่าง Agent กับ Environment ได้

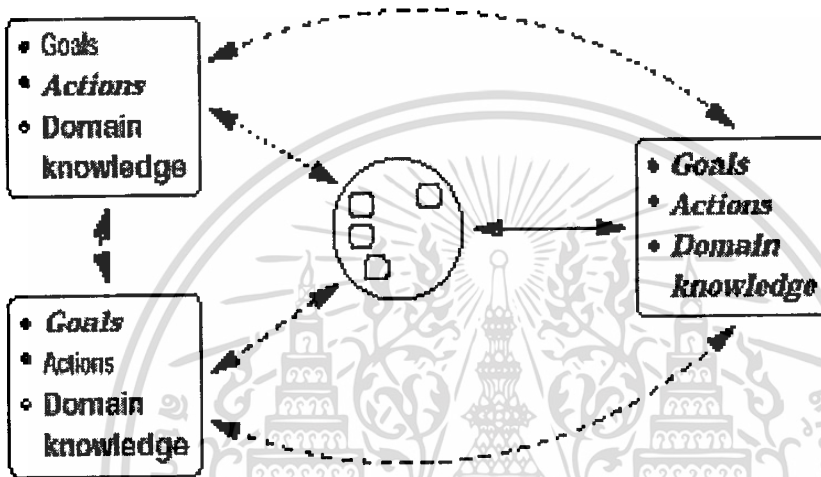


รูปที่ 2.3 โครงสร้างของ Homogeneous, Communicating MAS

ที่มา Layer Learning in Multi-Agent System

2.1.4 Heterogeneous, Communicating MAS

Agent ในรูปแบบนี้จะมีลักษณะขององค์ประกอบภายในที่แตกต่างกัน ทั้ง Goals , Actions , Domain knowledge จากรูปจะเห็นว่าภายในรูปสี่เหลี่ยมแต่ละรูปนั้นจะมีตัวอักษรที่ต่างกัน ซึ่งแสดงถึงความแตกต่างกันของ Agent และ Agent แต่ละตัวสามารถติดต่อสื่อสารถึงกันได้ รวมไปถึงสามารถที่จะทำการติดต่อสื่อสารกันระหว่าง Agent กับ Environment ได้



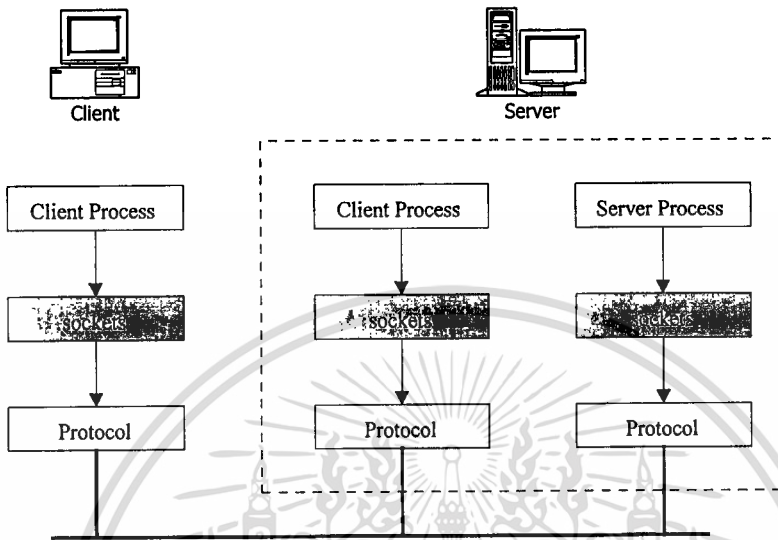
รูปที่ 2.4 โครงสร้างของ Heterogeneous, Communicating MAS
ที่มา Layer Learning in Multi-Agent System

2.2 กลไกในการติดต่อสื่อสาร

สำหรับ Robo Cup หรือ เกมจำลองการแข่งขันฟุตบอล โดยมีคอมพิวเตอร์เป็นผู้เล่น จัดเป็นการทำงานที่มีลักษณะแบบ Client/Server ซึ่งทำการติดต่อกันผ่านทาง UDP/IP Socket และอาจจะเปรียบเสมือนว่า 1 Agent นั้นแทนด้วย 1 Client

จากที่ได้กล่าวไปแล้วว่า Robo Cup ใช้ช่องทางการติดต่อกันผ่านทาง Socket ดังนั้น เราจึงจำเป็นที่จะต้องรู้วิธีการติดต่อสื่อสารผ่านทาง Socket เบื้องต้น เพื่อให้ Client และ Server สามารถติดต่อสื่อสารกันได้ ก่อนที่จะไปกำหนด Rule หรือต้องตกลงต่าง ๆ ต่อไป

Server ต้องสร้าง Socket เพื่อรอรับการเชื่อมต่อจาก Client และ Client ต้องสร้าง Socket เพื่อเชื่อมต่อกับ Server เช่นเดียวกัน โดยปกติแล้ว process บน server 1 process สามารถให้บริการกับ process บน client ได้หลาย process



รูปที่ 2.5 การติดต่อสื่อสารกันระหว่าง Client กับ Server ผ่านทาง Socket
 ที่มา Layer Learning in Multi-Agent System

2.2.1 Stream communication

วิธีนี้ มีความน่าเชื่อถือของข้อมูลมาก โดยจะทำการสร้าง Channel การเชื่อมต่อจนกระทั่งการติดต่อ นั้นเสร็จสิ้น (Connection-orient) ข้อมูลที่ถูกส่งไปจะได้รับการรับรอง ติดตามผลการส่ง ว่าส่งถึงหรือไม่ ถ้ามีปัญหา จะต้องทำการส่งใหม่ โดยใช้ Protocol TCP/IP ในการติดต่อ และมีการใช้ Function Read/Write เพื่อส่งและรับข้อมูลจาก Socket

Sockets used for streams

Requesting a connect ion

```

s = socket(AF_INET, SOCK_DGRAM, 0)
•
•
connect(s, ServerAddress)
•
•
write(s, "message", length)
•
•
    
```

Listening and acceptation a connection

```

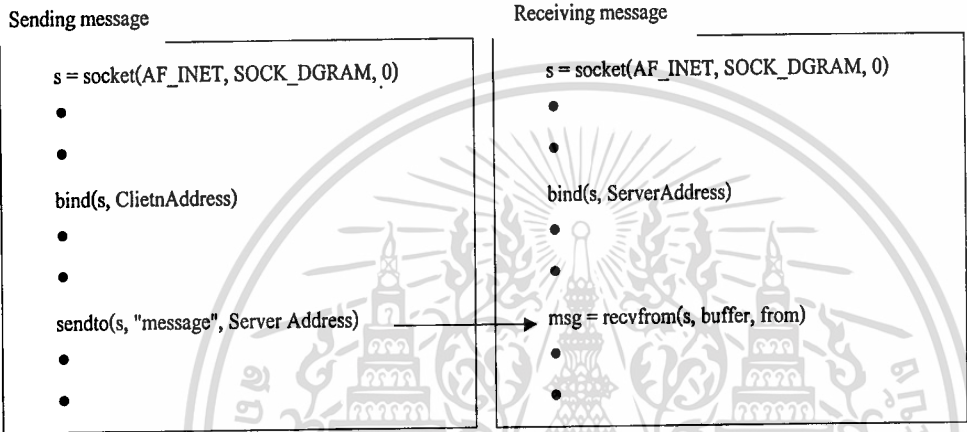
s = socket(AF_INET, SOCK_DGRAM, 0)
•
bind(s, ServerAddress)
listen(s, 5)
•
sNew = accept(s, from)
•
n = read(sNew, buffer, amount)
    
```

รูปที่ 2.6 Stream communication

2.2.2 Datagram communication

วิธีนี้ จะไม่สร้าง Channel ในการเชื่อมต่อ (Connection less) โดยที่เมื่อทำการส่งข้อมูลเสร็จ แล้วจะไม่มีการติดตามผลการส่งข้อมูล ไม่มีการตรวจสอบใดๆ ทั้งสิ้น โดยใช้ protocol UDP/IP ในการติดต่อ และมีการใช้ Function Send to/Recv from เพื่อส่งและรับข้อมูลจาก Socket

Sockets used for datagrams



รูปที่ 2.7 Datagram communication

หลังจากสร้าง Socket ขึ้นมาแล้ว การจะให้ Process ใช้งานได้ จะต้องทำการกำหนด Socket นั้นให้กับ Process ก่อน โดยกำหนดหมายเลข Port ที่จะใช้งานด้วย Function bind และปิดการเชื่อมต่อด้วย Function close

สำหรับ Robo Cup ข้อมูลที่จะใช้ในการส่งผ่าน Socket เป็นลักษณะข้อมูลที่เป็น Command และข้อมูลที่ลักษณะเป็น Information ต่างๆ ได้แก่ Connection command, Movement command, Control command, Sensory information ในลักษณะ Datagram communication ผ่านทาง UDP/IP Socket

2.3 หลักการและทฤษฎีที่นำมาใช้ในทีม FCPortugal

CMUnited99 และ FCPortugal ได้ถูกนำมาเป็นโปรแกรมต้นแบบ เพื่อใช้ในการทำงานของ การศึกษาและพัฒนา Agent ขึ้นมาใหม่ เนื่องจากจะช่วยทำให้ประหยัดเวลาในช่วงเริ่มแรกของการทำโครงการนี้ได้มาก และสามารถช่วยให้เกิดการวิจัยเฉพาะระบบ Multi-agent และการสื่อสารระหว่างโค้ดต่าง ๆ ซึ่งจากการศึกษาทีม FCPortugal นั้น ได้มีนำเสนอบทวิเคราะห์ต่างๆ ของทีมไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความยืดหยุ่นของทีมโดยการใช้แผนต่าง ๆ การวางรูปแบบทีม และการใช้ผู้เล่นให้เหมาะสมกับสถานการณ์ต่าง ๆ
- วัตถุประสงค์ของผู้เล่นได้ถูกแบ่งออกเป็น 3 ลักษณะ ได้แก่ การวางแผน, การรักษาบอลและการแย่งบอล (Strategic, Ball Possession, Ball Recovery)
- ความแตกต่างระหว่างการวางแผนล่วงหน้า และระหว่างการเล่น
- ความยืดหยุ่นในการเคลื่อนที่ไปตามตำแหน่งต่าง ๆ โดยใช้พื้นฐานของ utility function
- การติดต่อระหว่างผู้เล่นและ World state
- ระบบปัญญาประดิษฐ์ผ่านระบบการมองโดยใช้พื้นฐานของ Utility functions
- การเรียนรู้ของผู้เล่นในการวางตำแหน่ง ครอบบอล และการตัดสินใจ รวมไปถึงการตัดสินใจในการแย่งบอลคืน
- การเตะที่รุนแรงโดยใช้ระบบ Online optimization
- การรักษาประตูที่ชาญฉลาดในระบบสองมิติ (2D)
- เทคนิคการประกบตัวผู้เล่น
- การที่ Agent สามารถมองเห็น ได้ยิน รู้สึก คิด และแสดงออก
- ระบบตรวจสอบข้อผิดพลาดในระดับ World state

2.4 การวางแผนของทีม (Team Strategy)

เทคนิค, รูปแบบการเล่น, ตำแหน่ง และประเภทของผู้เล่นเป็นองค์ประกอบโดยรวมในฟุตบอล Robo Cup ทีมที่ดีจำเป็นต้องใช้องค์ประกอบเหล่านี้ในการเล่น ทีม CMUnited ได้นำเอารายละเอียดเหล่านี้ มาประยุกต์ใช้ โดยขึ้นอยู่กับผลและเวลาในการแข่งขันได้มีการทำการเพิ่มเติมรายละเอียดต่าง ๆ ลงไปในเทคนิค แผนและชนิดผู้เล่นใน แผนของทีม FC Portugal ขึ้นอยู่กับ เซตของประเภทหรือชนิดของผู้เล่น (Agent) และเซตของเทคนิคที่รวบรวมการวางตำแหน่งต่าง ๆ เข้าไว้ด้วยกัน การวางตำแหน่งผู้เล่นถูกใช้ในสถานการณ์ต่าง ๆ เช่น เป็นฝ่ายรับ รุก เปลี่ยนจากรับเป็นรุก การยิงจุดโทษ และอื่น ๆ

2.4.1 SBSP - Situation Based Strategic Positioning

หนึ่งในความแข็งแกร่งของ FC Portugal ก็คือการกำหนดตำแหน่ง การวางตัวลงตำแหน่งต่าง ๆ โดยใช้ความแตกต่างระหว่างการวางแผนก่อนการแข่งขัน และการเปลี่ยนแปลงแผนการเล่นในระหว่างการแข่งขัน SBSP (Situation Based Strategic Positioning) ใช้เพื่อการสร้างสถานการณ์ต่าง ๆ สำหรับสถานการณ์

ระหว่างการเล่น ตำแหน่งของ Agent ในสนามจะถูกกำหนด โดยการครอบครองบอล การแย่งบอล หรือการระบบการตัดสินใจในการหยุดเกม

การคำนวณตำแหน่งหรือจัดรูปแบบการลงแข่งขันนั้น Agent จะทำการวิเคราะห์ถึงเทคนิคและแผนการเล่นรวมถึงตำแหน่งของผู้เล่น Agent ในสนาม โดยตำแหน่งของผู้เล่นจะเปลี่ยนไปตามตำแหน่งของบอล ความเร็ว และสถานการณ์ เช่นการบุก การรับ การยิงประตู ซึ่งแผนของผู้เล่นนั้นได้รวมไปถึงการหาพื้นที่ในสนามที่กำหนดเส้น Offside และอื่น ๆ

2.4.2 ชนิดของผู้เล่น (Player Type)

ทีมฟุตบอลนั้นจะต้องประกอบไปด้วยผู้เล่นที่มีความสามารถต่างกัน และทำการเล่นในรูปแบบที่ต่างกัน ถึงแม้ว่า Simulated Agents จะทำหน้าที่แบบเดียวกันใน Soccer Server 6.06 ก็ตาม การที่มีความหลากหลายในการแสดงออกนั้นได้สร้างความยืดหยุ่นให้กับการเล่นได้ เช่น กองกลางสามารถที่จะบุกขึ้นไปข้างหน้าได้เช่นเดียวกับกองหน้า ในขณะที่ปีกสามารถที่จะเลี้ยงลูกได้ดี กองหลังมีการวางตำแหน่งผู้เล่นที่ดี กองหน้ามีการยิงประตูที่ดี เป็นต้น การที่ผู้เล่นแต่ละตำแหน่งมีความหลากหลายนั้น ได้ทำให้ความสามารถของทีม FC Portugal มีการพัฒนามากขึ้นกว่าเดิม

ลักษณะของผู้เล่นแต่ละตัวจะเป็นตัวแปรหลักใน SBSP สำหรับชนิดของผู้เล่น เช่น ผู้เล่นชนิดหนึ่งอาจจะชอบวิ่งในสนามเป็นระยะทางมาก ๆ ทำหน้าที่รับในบางครั้ง แต่ก็สามารถ วิ่งตามบอลได้ตามสถานการณ์ ผู้เล่นคนอื่น ๆ อาจใช้แผนอื่น เช่น รักษาตำแหน่ง และพักรักษาแรงของตัวเองเอาไว้ ลักษณะการครอบครองบอลเป็นปัจจัยหนึ่งของลักษณะผู้เล่น เช่น การส่งบอลแบบปลอดภัยไปให้ผู้เล่นที่เตรียมตัวในการรับบอล

2.4.3 DPRE - Dynamic Positioning and Role Exchange

DPRE (Dynamic Positioning and Role Exchange) จัดทำขึ้นโดยใช้พื้นฐานของงานจาก Perter Stone ที่เสนอความยืดหยุ่นในการเล่นของ Agent ด้วยโปรโตคอลสำหรับการเปลี่ยนแผนการในการแข่งขันนั้น ๆ ใน FCPortugal ผู้เล่น (Agent) จะมีการสลับทั้งตำแหน่ง และประเภทของผู้เล่นในแผนนั้น ๆ ด้วย โดยที่การเปลี่ยนตำแหน่งจะกระทำก็ต่อเมื่อการกระทำนั้นสร้างผลดีให้กับทีมที่กำหนดตำแหน่งนั้นต้องขึ้นอยู่กับแผนการเล่นและตำแหน่งของบอล ยกตัวอย่าง เช่น ทีมที่มีสองแผนการเล่น รับ และ รุก ได้มีการกำหนดกองหลังตัวกลางให้มีความสำคัญในการรักษาแผงกองหลัง ในขณะที่บุกความสำคัญของกองหน้าก็จะเพิ่มขึ้น ถึงแม้ว่าการเปลี่ยนตำแหน่งจะทำสำเร็จแล้วก็ตาม การเปลี่ยนตำแหน่งเพิ่มอีกครั้ง และการรักษาตำแหน่งก็สามารถกระทำได้ แต่ไม่ได้ถูกนำมาใช้ใน RoboCup2000 จากการวิจัยพบว่า DPRE (Dynamic

Positioning and Role Exchange) ได้ช่วยให้สภาพโดยรวมของทีมดีขึ้น ซึ่งทีม FCPortugal ที่ได้นำเอา DPRE มาใช้นั้น ชนะถึง 85 เปอร์เซ็นต์ เมื่อเทียบกับการที่ไม่ได้นำเอา DPRE (Dynamic Positioning and Role Exchange) มาใช้ ยิ่งไปกว่านั้นยังสามารถทำให้ทีมเล่นได้ดีถึงแม้จะมีผู้เล่นเพียง 6-7 คนก็ตาม

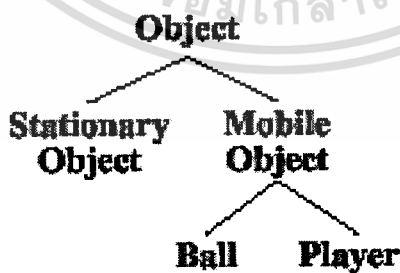
2.5 Agent Architecture and High Level Decision Module

FCPortugal เป็นทีมที่รวบรวมความรู้ทางด้านเทคนิคการเล่น การวางตำแหน่ง และชนิดผู้เล่นเข้าด้วยกัน องค์ประกอบเหล่านี้ถูกกำหนดโดยโค้ชโดยแสดงถึงลักษณะของทีมเป็นสำคัญ ตัวควบคุมหลักจะทำหน้าที่วิเคราะห์ และพยากรณ์เพื่ออัปเดต World State โมเดลใน FC Portugal แบ่งเป็น 4 ระดับด้วยกัน

- Global situation Information ผลการแข่งขัน เวลา สถิติของเกม และ การแสดงออกขององค์ประกอบต่าง ๆ เหมาะสำหรับการนำมาวิเคราะห์เทคนิคของทีมในช่วงเวลาที่กำหนด
- Situation Information ข้อมูลที่เกี่ยวข้องกับ SBSP, SLM และ ADVCOM
- Action Selection Information เซตของตัวแปรต่าง ๆ เพื่อใช้ในการแสดงถึงสถานการณ์ที่กำลังเกิดขึ้นเพื่อเลือกระหว่างการครอบครองบอลหรือการแย่งบอล
- World State ข้อมูลทั่วไป รวมถึงตำแหน่งและความเร็วของผู้เล่นและลูกบอล

2.5.1 World State

เป็นสภาวะแวดล้อมที่เกิดขึ้นในขณะที่ทำการแข่งขัน สำหรับการกำหนดการกระทำใด ๆ ณ เวลาใดเวลาหนึ่ง จำเป็นที่จะต้องรู้ถึงองค์ประกอบนั้น ๆ ด้วย เพื่อให้สามารถกำหนดการกระทำที่จะต้องแสดง ณ เวลานั้นได้



รูปที่ 2.8 Objects type hierarchy

ที่มา Layer Learning in Multi-Agent System

จากรูปที่ 2.7 องค์ประกอบต่างๆ ใน world state ประกอบด้วย object ต่าง ๆ ดังนี้

- stationary object

เป็น object ที่หยุดนิ่ง ไม่มีการเคลื่อนที่ใด ๆ ได้แก่ goals, sideline และ field markers ต่าง ๆ

- mobile object

เป็น object ที่มีการเคลื่อนที่ และเปลี่ยนแปลงตำแหน่งอยู่ตลอดเวลา ได้แก่ ผู้เล่นเอง (agent), ลูกบอล (ball) และผู้เล่นอื่น ๆ (player) ซึ่งประกอบด้วยผู้เล่นในทีม (teammate) 10 คนและผู้เล่นฝ่ายตรงกันข้าม (opponent) อีก 11 คน

ซึ่ง object ทั้งหมด สามารถแสดงในลักษณะลำดับชั้นได้ ดังรูปที่ 2.7

ณ เวลาใด ๆ ผู้เล่น (agent) จะต้องเก็บข้อมูลของตำแหน่ง object เหล่านี้ ซึ่งได้แก่ goal, sideline, field marker, ball และ player ทั้ง 21 คน ตามที่ผู้เล่นสามารถเห็นได้ ณ เวลานั้น เพื่อเป็นตัวกำหนดทิศทางการเคลื่อนที่ และการกระทำของผู้เล่นเองต่อไป ซึ่งผู้เล่นสามารถหาข้อมูลที่เกี่ยวข้องได้จาก

- Visual information จาก Soccer server
- Audial information จาก Soccer server
- Sense_body information จาก Soccer server
- Predicted effects การคาดเดาล่วงหน้า จากเหตุการณ์ก่อนหน้านี้ โดยผู้เล่นเอง

โดยที่ visual information จะให้ข้อมูลในระยะทาง และมุมระหว่างผู้เล่นกับ object อื่นเท่าที่ผู้เล่นสามารถจะเห็นได้ , audial information เป็นข้อมูลที่ผู้เล่นอื่นบอกมา เพราะว่าผู้เล่นไม่สามารถมองเห็น object ทั้งหมดได้ ในตำแหน่ง และทิศทางที่ต่างกัน ทำให้ object ที่ผู้เล่นแต่ละคนมองเห็นจึงต่างกัน แต่ระหว่างผู้เล่น (Agent) นั้นสามารถจะสื่อสารกันได้ เพื่อรักษาข้อมูลต่าง ๆ ให้คงอยู่ , sense_body เป็นข้อมูลที่กำหนดขอบเขตของมุมมองของผู้เล่นเอง และสุดท้าย Predicted effects เป็นการคาดเดาจากผู้เล่นเอง โดยอาศัยข้อมูลที่มีอยู่ก่อนหน้านี้ ซึ่งอาจรู้ทิศทางการเคลื่อนที่ของ object ต่าง ๆ และคาดเดาได้

2.5.2 Action

เป็นการกระทำที่จะเกิดขึ้นในสถานการณ์หนึ่ง ๆ ซึ่งสามารถแบ่งออกได้ใน 2 ลักษณะ คือ basic action และ advance action

2.5.2.1 Basic action

เป็นการกระทำที่เกิดขึ้นจากคำสั่งพื้นฐานของคำสั่งประเภท movement command ซึ่งมีดังนี้

- Turn คือการหมุนตัวเอง

- Dash คือการวิ่งไปข้างหน้าและเตะบอล
- Kick คือการเตะบอล
- Move คือการเคลื่อนตัวเอง
- Catch คือการรับบอล

2.5.2.2 Advance action

เป็นการกระทำที่เกิดขึ้นต่อเนื่องกัน โดยอาศัย basic action ทำงานประสานกัน เพื่อทำให้เกิด action ต่าง ๆ ได้ดังนี้

- Dribbling คือ การเลี้ยงบอล โดยอาศัยคำสั่ง dash, turn และ kick เพื่อประคองให้ลูกบอลอยู่กับตัวผู้เล่นตลอดระยะทางการเลี้ยงบอลนั้น
- Shotting คือ การยิงประตู เป็นการ kick บอล เพื่อยิงเข้าประตูผู้เล่นฝ่ายตรงข้าม ผู้ยิงประตู ควรเตะด้วยแรงที่มากที่สุดที่สามารถทำได้ เพื่อให้บอลเคลื่อนที่ได้เร็ว และตรงเป้าหมายมากที่สุด
- Passing คือ การส่งบอล เป็นการ kick บอลเพื่อส่งให้เพื่อนร่วมทีม (teammate) ซึ่งผู้ส่งควรเตะให้เบาว่าการยิงประตู เพราะผู้เล่นจะสามารถรับบอลได้ง่ายกว่า และโอกาสที่ผู้รับจะพลาดน้อยกว่า
- Interception คือ การสกัดบอล เป็นการ turn, dash และ kick เพื่อให้ผู้เล่นที่อยู่ใกล้เส้นทางที่บอลวิ่งผ่าน สามารถเข้าไปถึงบอลได้ก่อนที่ฝ่ายตรงข้ามจะเข้ามารับบอล
- Clearing คือ การ clear บอล เมื่อมีบอลเข้ามาบริเวณหน้าประตูตนเอง ถ้าผู้เล่นไม่สามารถส่งบอล หรือเลี้ยงบอลออกไปได้ ผู้เล่นควรเลือกที่จะเตะทิ้ง คือเตะบอลออกไปให้อยู่ไกลบริเวณหน้าประตูตนเองมากที่สุด

2.6 Intelligent Perception and Communication

FC Portugal low-level world state model นั้นสร้างขึ้นจากพื้นฐานของ world state model จาก CMUUnited99 Source Code โมเดลนี้ได้ทำการบันทึกตำแหน่งและความเร็วของวัตถุในสนาม ตำแหน่งต่าง ๆ ถูกกำหนดไว้ด้วยค่าสูงสุด 1.0 เมื่อผู้เล่นมองวัตถุต่าง ๆ เขาสามารถกำหนดตำแหน่งต่าง ๆ ได้จากระยะทางระหว่างวัตถุนั้นกับตัวผู้เล่น เช่นเดียวกับการบันทึกค่าความเร็ว ทั้งสองส่วนนี้มีประโยชน์อย่างยิ่งเมื่อผู้เล่นได้ทำการสื่อสารข้อมูลต่าง ๆ กับ world state knowledge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 ADVCOM Intelligent Communication Mechanism

เมื่อมีการสื่อสารระหว่างกันของผู้เล่นของทีม FC Portugal โดยส่งข้อมูลต่าง ๆ ออกมา เช่น เทคนิคของผู้เล่น ความเร็วบอล และอื่น ๆ เพื่อคำนวณ และตัดสินใจว่าควรจะเริ่มการส่งถ่ายข้อมูลหรือไม่ ซึ่งการสื่อสารจะเกิดขึ้นเมื่อผู้เล่นเห็นว่าข้อมูลของตนนั้นมีประโยชน์มากกว่าของผู้ร่วมทีม

2.6.2 SLM Strategic Looking Mechanism

ในโปรแกรมที่ซับซ้อนอย่าง เช่น ฟุตบอล ตัวเซนเซอร์จำเป็นต้องถูกนำมาเป็นส่วนประกอบเพื่อช่วยให้ agents สามารถทำงานได้เต็มประสิทธิภาพ อย่างไรก็ตามความสำคัญของสภาพแวดล้อมก็เป็นส่วนสำคัญในการตัดสินใจในการทำงานของตัวผู้เล่นนั้น ๆ เช่น ถ้าหากบอลอยู่ใกล้กับโกล การที่จะมองการเคลื่อนไหวของโกลย่อมสำคัญกว่าการมองผู้เล่นอื่น ๆ

FC Portugal SLM intelligent นั้นมีระบบในการช่วยการตัดสินใจในการหาทิศทางของตัวผู้เล่นโดยใช้องค์ประกอบของชนิดผู้เล่น การวางตำแหน่ง ตำแหน่งของผู้เล่น และความมั่นใจในการดำเนินเกมของผู้เล่น รวมไปถึงการคำนวณการเคลื่อนที่ของวัตถุต่าง ๆ ในสนาม ซึ่งในการคำนวณนี้จะให้คำตอบที่ขึ้นอยู่กับสถานการณ์ โดยการให้ความสำคัญในการมองวัตถุโดยดูว่าสิ่งไหนควรมองก่อน แล้วจึงส่งคำสั่งในการหันไปมองวัตถุนั้นตามแต่เห็นสมควร

2.7 Integrating Soccer Knowledge

ผู้เล่นแต่ละคนจะมีการตัดสินใจที่เป็นอิสระจากกัน โดยการใช้ส่วนที่ได้เรียนรู้มาแล้ว โดยแสดงผ่านภาพสองมิติ ส่วนที่เรียนรู้นี้รวมไปถึงแบบแผนการตัดสินใจในขณะที่ครองบอล และขณะที่ตัวผู้เล่นเป็นอิสระ รวมไปถึงขณะที่เกมหยุดลง โดยใช้พื้นฐานที่สมจริง

2.7.1 Ball Possession

ถ้าผู้เล่นครองบอลอยู่ การตัดสินใจจะมีดังนี้

- Shoot ทำการยิงประตู โดยจะยิงแรง หรือเร็วขึ้นขึ้นอยู่กับโอกาสในการทำประตู
- Pass การส่งบอล โดยรวมจะเป็นการผ่านบอลเร็ว โดยส่งไปให้ใกล้ตัวผู้เล่นอีกคนหนึ่งให้มากที่สุด
- Forward การส่งบอลไปข้างหน้าผู้เล่นทีมเดียวกัน เพื่อให้ผู้เล่นนั้นวิ่งไปรับลูก คำนวณโดยการให้จุดที่รับลูกเป็นจุดที่บอลหยุดนิ่งแล้ว ต่างจากการส่งธรรมดาที่บอลจะไม่หยุดจนกว่าจะถึงตัวผู้เล่นในทีมเดียวกัน
- Dribble การเลี้ยงลูกไปยังจุดที่กำหนดไว้
- Hold การรักษาบอลไว้กับตัวผู้เล่นเองโดยไม่เคลื่อนที่ไปไหน

ในการเลือกว่าจะส่ง ยิง เลี้ยงหรือครองบอลไว้ ขึ้นอยู่กับว่าผลลัพธ์ที่ออกมาอันไหนจะดีที่สุด โดยมีการถ่วงน้ำหนักให้กับกรกระทำต่าง ๆ เช่น การส่ง จะรวมการคำนวณสถานการณ์ไว้ถึง 12 ส่วน (ตำแหน่ง, จุดที่จะส่ง, ความน่าจะเป็นที่จะถูกตัดลูกไปได้ และอื่น ๆ) การให้ Scale ที่ต่างกันและการถ่วงน้ำหนักจะถูกนำมาคำนวณด้วย โดยทั่วไปแล้วการส่งกลับไปยังแนวหลังจะไม่ถูกนำมาคิด

2.7.2 Ball Recovery

การแย่งบอลคืนนั้นทำได้ในหลายสถานการณ์ เช่น การคุมตัวผู้เล่นอีกฝ่ายหนึ่ง หรือการรักษาประตู ในการแย่งบอลคืนทำได้โดย

- Ball Interception การวิ่งไปยังจุดที่สามารถแย่งบอลได้ให้เร็วที่สุด
- Ball Passive Interception การตัดบอล ไม่ใช่จุดที่จะสามารถแย่งบอลได้ แต่เป็นจุดที่ได้เปรียบมากกว่า ถึงแม้จะใช้เวลามากก็แต่ก็ทำให้สามารถเปลี่ยนผู้ครองบอลได้
- Mark Pass Line การสกัดการส่งบอล โดยการขวางทางการส่ง โดยจะกระทำเมื่อผู้เล่นสามารถตรวจพบจุดที่เหมาะสมในการขวางทางการส่งไปยังผู้เล่นอีกฝั่งที่ว่างอยู่
- Approach Ball Position ถึงแม้จะไม่สามารถแย่งบอลมาได้แต่ก็เป็นการบีบผู้เล่นอีกฝ่ายไม่ให้ทำอะไรได้มากนัก
- Mark Opponent การประกบตัวผู้เล่นเพื่อไม่ให้โอกาสได้เป็นฝ่ายได้เปรียบ
- Cover Goal ผู้เล่นทำการรักษาประตูฝ่ายตัวเองทำหน้าที่ตั้งรับ

2.7.3 Goalie Defensive Strategy

การทำงานของผู้รักษาประตุนั้นจะสมจริงเมื่อมีการคำนวณในแบบสามมิติ ในการคำนวณแบบสองมิตินั้นเราได้ตัดการยิงบอลข้ามคานประตูออกไป จึงเสมือนการยิงจุดโทษเสียมากกว่า ซึ่งเราได้อธิบายลักษณะของผู้รักษาประตูไว้ดังนี้

- Mark Position ผู้รักษาประตูสามารถเดินในเขตสี่เหลี่ยมของจุดโทษเท่านั้น เมื่อบอลเคลื่อนที่เข้ามา ๆ การจัดตำแหน่งจะอยู่ภายในกรอบสี่เหลี่ยมจุดโทษ โดยใช้จุดตัดระหว่างเส้นทางผ่านบอลและตำแหน่งในกรอบนั้นบางครั้งเมื่อไม่มีผู้ใดครองบอลและบอลยังเคลื่อนที่ด้วยความเร็วอยู่ ผู้รักษาประตูต้องคำนวณถึงจุดที่จะสามารถเข้าไปรับบอลได้ แทนที่จะเป็นการวิ่งไล่ตามบอล
- Passive Interception จะกระทำเมื่อสามารถรับบอลได้โดยง่าย โดยจะรับบอลก็ต่อเมื่อผู้รักษาประตูสามารถไปยังจุดที่จะตัดบอลได้ก่อนที่ผู้เล่นอื่นจะได้บอล

- Active Interception โดยทั่วไปแล้ว ผู้รักษาประตูต้องวิ่งไปยังบอลเพื่อที่จะรับบอลก่อนที่ผู้เล่นอีกฝ่ายจะทำการยิงประตู
- Goal Interception เป็นทางเลือกสุดท้าย เมื่อผู้รักษาประตูคิดว่าไม่สามารถดับบอลด้วยวิธีอื่นได้ และบอลเคลื่อนที่มายังทางประตู ผู้รักษาประตูจะพยายามรับบอลโดยคำนวณจากเส้นประตูและเส้นทางของบอลมายังประตู
- Catch จะทำการจับบอลเมื่อมีโอกาส โดยปกติแล้วข้อมูลนี้จะมีข้อผิดพลาดอยู่เมื่อผู้รักษาประตูคิดว่าสามารถจับลูกได้ในขณะที่สถานการณ์จริง ๆ แล้วไม่เป็นเช่นนั้น

2.7.4 Low-Level Skills – Optimization Kick

เนื่องจาก FC Portugal ไม่เน้น low-level skills และ Source Code นั้นสามารถหาได้จากที่ต่าง ๆ บนอินเทอร์เน็ต ดังนั้นความสามารถของโปรแกรมส่วนใหญ่จึงได้ถูกปรับปรุงจาก CMUnited99 อย่างไรก็ตาม FC Portugal ได้พัฒนาการเตะโดยใช้พื้นฐานของ Optimization techniques รวมไปถึงการเลี้ยงบอล การป้องกัน และการรักษาประตูด้วย

ความสำเร็จของผู้เล่นหรือ Agent นั้น จะขึ้นอยู่กับความสามารถในการเล่น ซึ่งความสามารถที่สำคัญที่สุดได้แก่ การเตะแรงในทุกสถานการณ์ โดยการใช้การวิเคราะห์ข้อมูลต่าง ๆ และทฤษฎีการเรียนรู้ Learning techniques เราได้สุ่มปัญหาขึ้นมาและใช้ simple search algorithm เพื่อกำหนดจังหวะในการเตะใน Soccer server นั้น ได้มีการทำนายตำแหน่งและความเร็วของบอลไว้ก่อน และจึงมาหาวิธีการเตะ ซึ่งจะได้แบ่งเป็นสองส่วนดังนี้ คือ ส่วนแรก ทำการสุ่มเพื่อหาวิธีที่ดีที่สุด ส่วนที่สอง ก็จะพยายามเพิ่มความสามารถของวิธีการที่สุ่มได้ให้ดีขึ้นไปอีก เทคนิคที่ใช้ในการสุ่มนั้นทำโดยการสุ่มเตะไปเรื่อย ๆ จนกว่าบอลนั้นจะออกจากตำแหน่งที่สามารถเตะได้ การเตะครั้งสุดท้ายจะถูกแทนที่ด้วยค่าที่วิเคราะห์ขึ้นมา และค่าที่ได้จะถูกเก็บรวบรวมไว้เพื่อหาความเร็วสูงสุดในการเตะด้วยวิธีต่าง ๆ ผลที่ได้จากการที่ Agent เตะบอลที่อยู่ตรงหน้าให้ผลเป็นที่น่าพอใจ ในความเร็วระดับปานกลาง จากการทดสอบกว่า 2000 ครั้ง ในความเร็วระดับ 2.25 จำนวน 500 รูปแบบ FC Portugal จะเลือกตัวเลข 500 เนื่องจากลดเวลาในการคำนวณลง หลังจากการค้นหาแบบสุ่ม วิธีที่ดีที่สุดในการเตะได้ถูกพัฒนาโดยใช้ hill climbing algorithm เพื่อหาจุดที่ให้ความเร็วสูงสุดในการเตะบอลนั้น

บทที่ 3

การทำงานของ Soccer Server

Soccer Server นั้นเป็นระบบที่อนุญาตให้ Agent หรือ Client ที่ทำหน้าที่แทนผู้เล่นฟุตบอล เชื่อมต่อเข้ามาผ่านทาง Network โดยจะการติดต่อระหว่าง Client กับ Server ผ่านทาง UDP Port 6000 และสร้าง Socket เพื่อเป็นช่องทางการติดต่อสื่อสาร โดย Soccer Server จะประกอบไปด้วย 2 ส่วน คือ Soccer server กับ Soccer monitor

3.1 Soccer Server

Soccer Server คือ โปรแกรมที่ทำหน้าที่ให้บริการการจำลองการแข่งขันของฟุตบอลลงมาเล่นบนเครื่องคอมพิวเตอร์ ซึ่งจะทำหน้าที่แสดงการจำลองการเล่นที่ของ ลูกบอล ผู้เล่น และการติดต่อสื่อสารกันระหว่าง Agent กับ Soccer server รวมไปถึงการควบคุมให้เกมส์การแข่งขัน เป็นไปตาม rules ที่ตั้งไว้

3.2 Soccer monitor

Soccer monitor คือ โปรแกรมที่ทำหน้าที่แสดงภาพการแข่งขันที่เกิดขึ้นจาก Soccer server ให้ออกมาเป็นภาพ 2 มิติ บนสนามจำลองที่อยู่บนหน้าต่างของ ระบบ X-windows

3.3 การเรียกใช้งาน Soccer Server

ในการพัฒนาโครงการนี้จะใช้ โปรแกรม Soccer Server version 7.08 โดยในขั้นตอนแรกเราจะต้องทำการ Download โปรแกรม Soccer Server มาใช้งานก่อน จาก <http://www.robocup.org> เมื่อเราได้ตัวโปรแกรมแล้วก็จะนำมาทำการ Unzip ด้วยคำสั่ง

```
$ gzip -dc sserver-*.tar.gz
```

- หลังจาก Unzip ข้อมูลแล้ว จะต้องทำการสั่ง Configuration ให้กับ soccer server ด้วยคำสั่ง

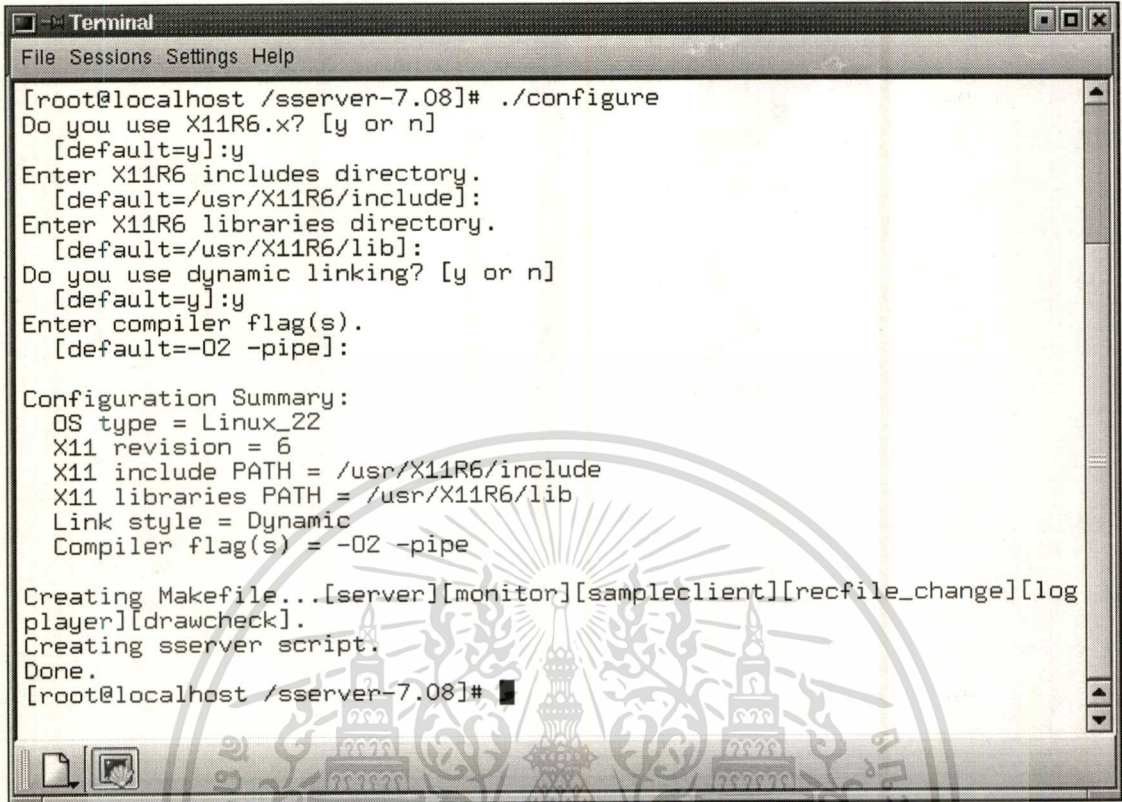
```
$ configure
```

- ทำการสั่ง make เพื่อทำการ Compile โปรแกรม soccer server

```
$ make
```

- ทำการสั่งคำสั่ง sserver เพื่อเรียกใช้งาน Soccer Server

```
$ sserver
```



```

Terminal
File Sessions Settings Help

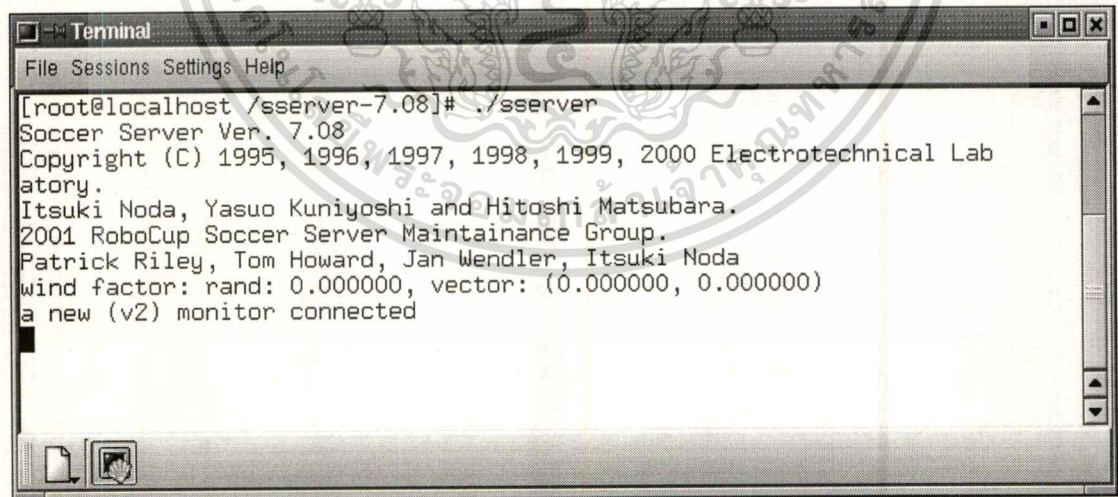
[root@localhost /sserver-7.08]# ./configure
Do you use X11R6.x? [y or n]
[default=y]:y
Enter X11R6 includes directory.
[default=/usr/X11R6/include]:
Enter X11R6 libraries directory.
[default=/usr/X11R6/lib]:
Do you use dynamic linking? [y or n]
[default=y]:y
Enter compiler flag(s).
[default=-O2 -pipe]:

Configuration Summary:
OS type = Linux_22
X11 revision = 6
X11 include PATH = /usr/X11R6/include
X11 libraries PATH = /usr/X11R6/lib
Link style = Dynamic
Compiler flag(s) = -O2 -pipe

Creating Makefile...[server][monitor][sampleclient][recfile_change][log
player][drawcheck].
Creating sserver script.
Done.
[root@localhost /sserver-7.08]#

```

รูปที่ 3.1 การใช้คำสั่ง configure



```

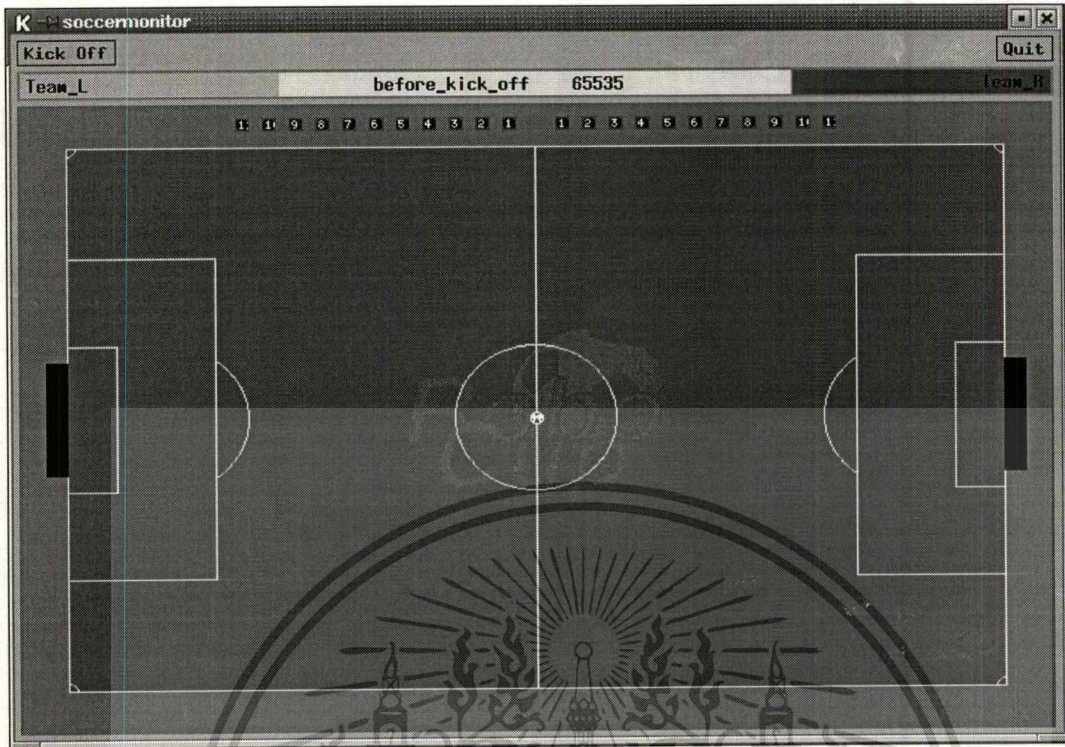
Terminal
File Sessions Settings Help

[root@localhost /sserver-7.08]# ./sserver
Soccer Server Ver. 7.08
Copyright (C) 1995, 1996, 1997, 1998, 1999, 2000 Electrotechnical Lab
atory.
Itsuki Noda, Yasuo Kuniyoshi and Hitoshi Matsubara.
2001 RoboCup Soccer Server Maintenance Group.
Patrick Riley, Tom Howard, Jan Wendler, Itsuki Noda
wind factor: rand: 0.000000, vector: (0.000000, 0.000000)
a new (v2) monitor connected

```

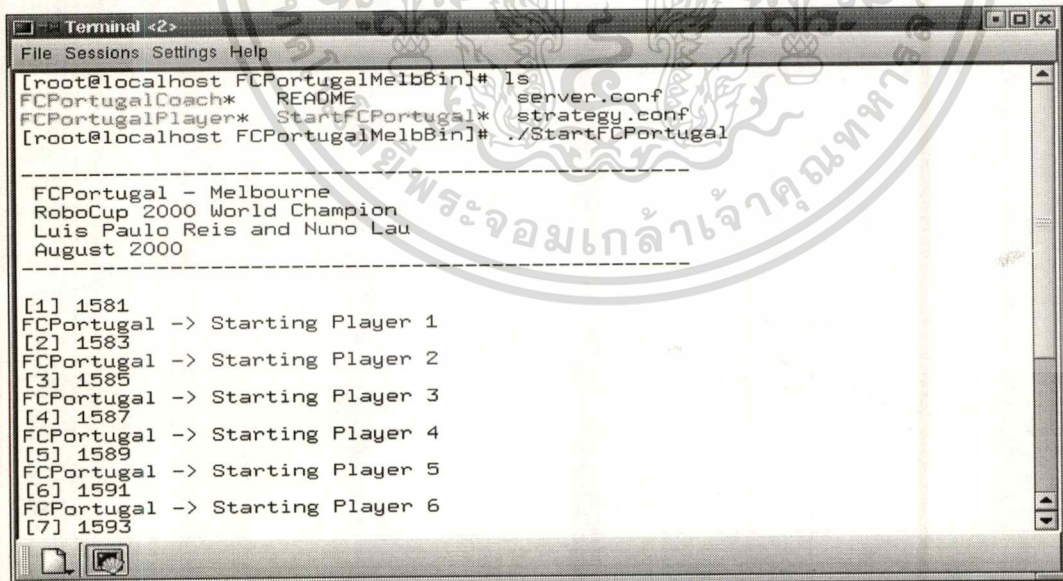
รูปที่ 3.2 การใช้คำสั่ง sserver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 – สนามที่สร้างจาก Soccer Monitor

- หลังจากทำการตั้งค่าสั่ง sserver แล้วจะได้ภาพสนามฟุตบอลดังรูปที่ 3.3
- จากนั้นจะทำการเรียก Agent ของทีม CMUnited กับทีม FC Portugal เพื่อทำการเชื่อมต่อเข้ากับ Soccer server ด้วยคำสั่ง Start ของแต่ละทีม



รูปที่ 3.4 การใช้คำสั่งให้ FCPortugal ลงสนามด้วยคำสั่ง StartFCPortugal

```

Terminal <2>
File Sessions Settings Help

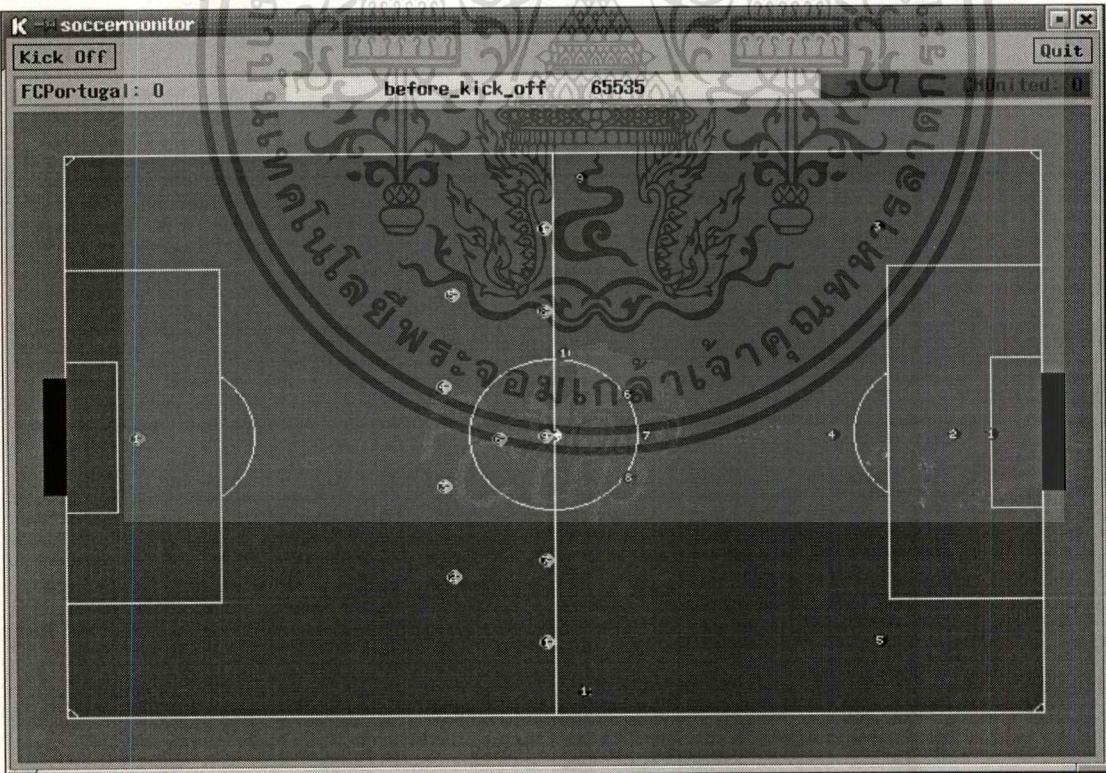
[root@localhost /project]# cd CMUnited99-linux
[root@localhost CMUnited99-linux]# ls
CMUnited-98-sim.ps.Z  CMUnited99*      client.conf      dt-data.tree
CMUnited-99-sim.ps.Z  FCPortugalCoach*  core             server.conf
CMUnited-99-sim.ps.gz  README           dt-data.names   start*
[root@localhost CMUnited99-linux]# ./start

-----
CMUnited99 was created by Peter Stone, Patrick Riley, and Manuela Veloso
Copyright 1999. All rights reserved.
-----

Running with the following parameters
Hostname: localhost
Team: CMUnited
Program: ./CMUnited99
[1] 1615
./dt-data
[2] 1617
./dt-data
[3] 1619
./dt-data
[4] 1621
./dt-data
[5] 1623

```

รูปที่ 3.5 การใช้คำสั่งให้ CMUnited99 ลงสนามด้วยคำสั่ง Start



รูปที่ 3.6 ภาพหลังจากการใช้คำสั่งลงสนามของทั้ง 2 ทีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 Sensor Information

กลุ่มคำสั่งชุดนี้จะเป็นคำสั่งที่ทำหน้าที่เกี่ยวกับการรับรู้ถึงข้อมูลในการแข่งขัน โดย Server จะทำการส่งข้อมูลเหล่านี้ให้ Client หรือ Agent

(hear Time Direction Message)

เป็นคำสั่งที่รับการประกาศมาจาก Server ในกรณีที่มี Client เรียกใช้คำสั่ง say Message ไปขอข้อมูลจาก Server และ Server จะทำการตอบกลับออกไปยัง Client ทุกตัว โดยที่ Direction จะเป็นค่าของผู้ส่ง (Sender)

(see Time ObjInfo ObjInfo ...)

เป็นคำสั่งที่ Server จะทำการส่งข้อมูลต่าง ๆ ของแต่ละ Object ไปยัง Client ObjInfo ประกอบด้วย ObjName , Distance , Direction , DistChng , DirChng , FaceDir ObjName ประกอบด้วย (player Teamname Unum) , (goal Side) , (ball) , (flag) , (line)

โดยปกติแล้ว Server จะทำการส่งข้อมูลออกมาเป็นในลักษณะ Boardcast ตลอดเวลา แล้วแต่จะให้ ค่าของ config Time เป็นอย่างไร และ Client ก็จะคอยรับฟังข้อมูลตลอดเวลาเช่นกันตามจังหวะเวลาที่ Server ส่งมาให้ โดยถ้าทาง Server เป็นฝ่ายส่งไปให้ Client แล้ว Client รับข้อมูลเก็บไว้ ทางฝั่ง Client ก็จะมี Messages ว่า “recv No.” (No คือ หมายเลขของ Messages ที่ได้รับมา) ส่วนทางด้าน SERVER ที่รับคำสั่งที่ส่งไปจาก Client ก็จะมี Messages แสดงที่ Soccer Monitor ด้านล่างว่า “Send Command” (Command คือ คำสั่งที่ส่งมาจาก Client เช่น Move X Y เป็นต้น)

ตัวอย่างของ Messages Sensor Information ที่ Server Boardcast ออกไป และ Client ทำการรับข้อมูลมาจาก Server

recv 2035 : (see 0 ((goal r) 73 -7) ((flag r t) 84.8 -31) ((flag r b) 76.7 18) ((ball) 22.2 -26 0 0)

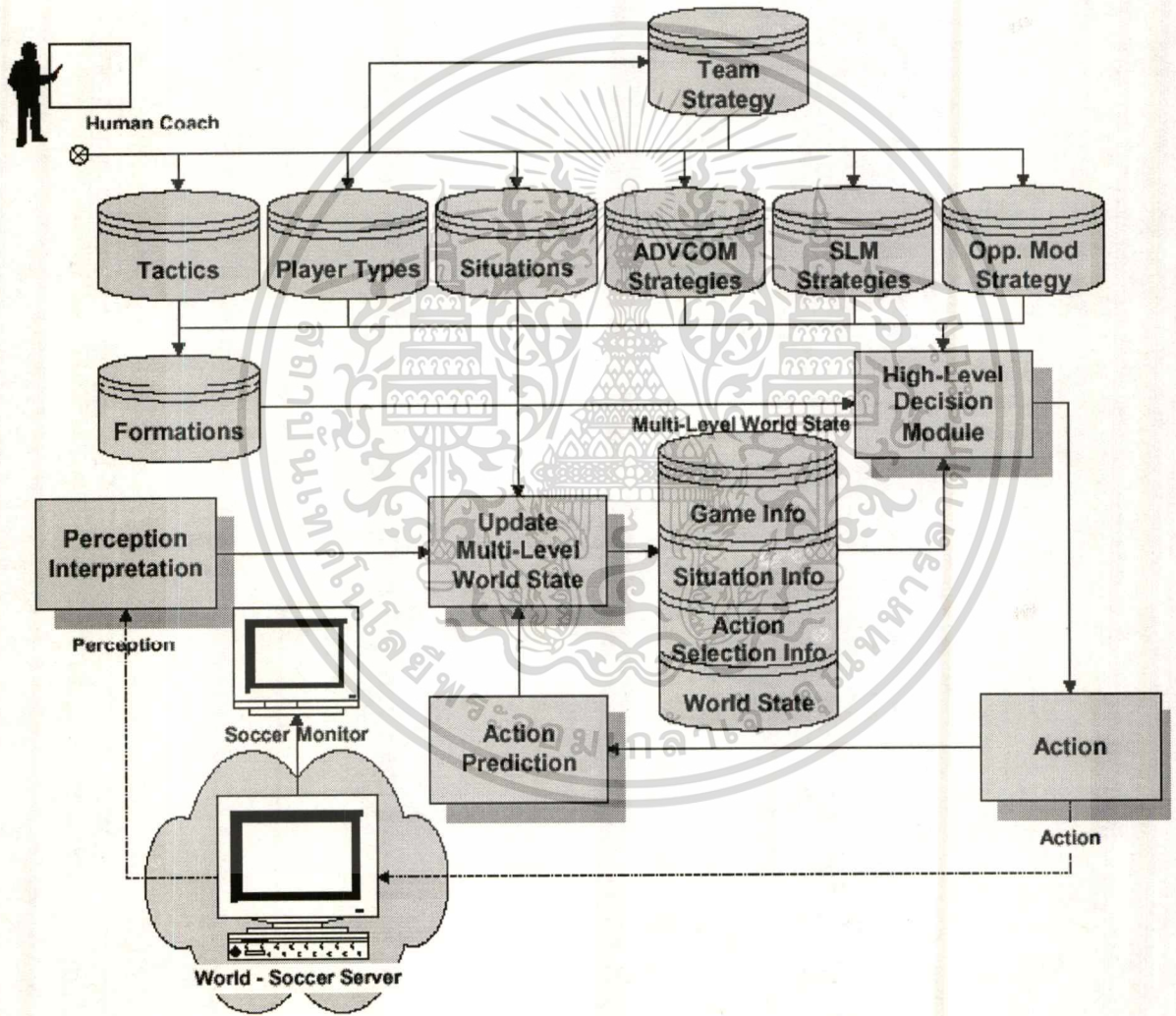
- อธิบาย Sensor Information ลำดับที่ recv 2035

ที่เวลา 0 (อาจจะเป็น before_kick_off) ที่ด้านขวาของ goal (goal r) เป็นระยะ 73m และ ทำมุม 7 degrees ไปทางซ้าย และมุมตรงด้านขวาบน (flag r t) ห่างเป็นระยะ 84.8m และทำมุม 31 degrees ไปทางซ้าย มุมตรงด้านขวาล่าง (flag r b) ห่างเป็นระยะ 76.7 m ทำมุม 18 degrees ไปทางขวา และลูกบอล (ball) อยู่ห่าง 22.2m ทำมุม 26 degrees ไปทางซ้าย และมีความเร็วเป็น 0 เป็นต้น

บทที่ 4

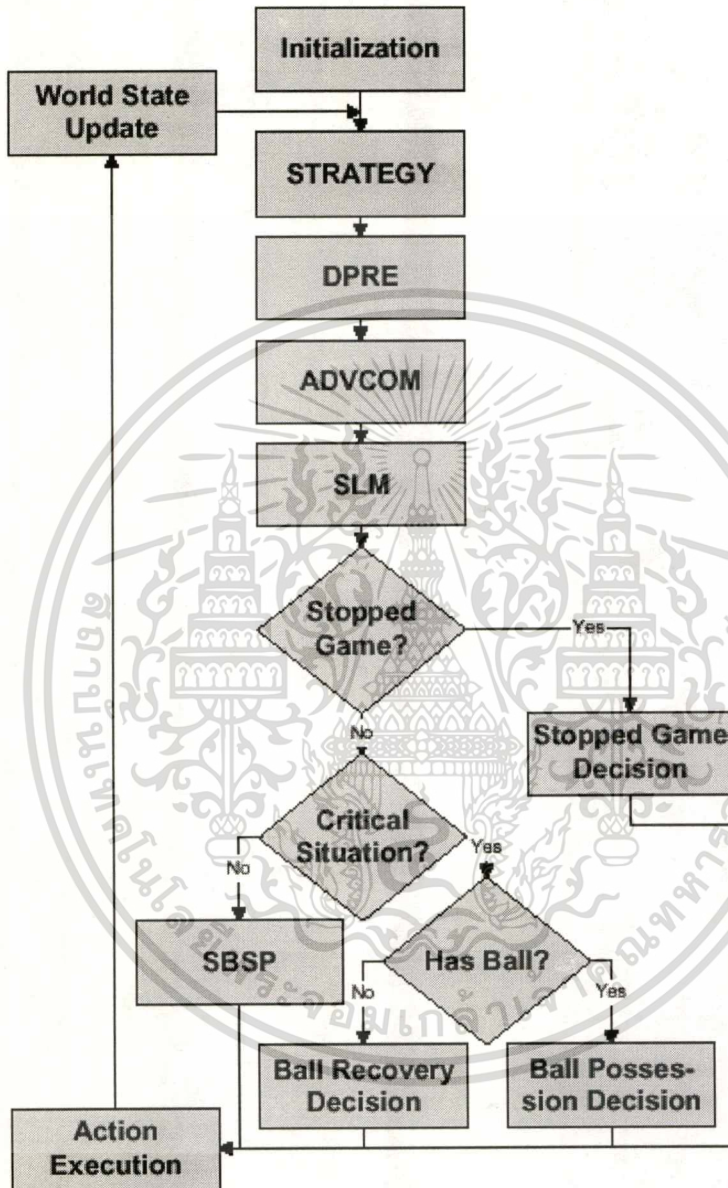
การพัฒนา Agent Team

4.1 Agent Architecture and Control Flow of the High Level Decision Module



รูปที่ 4.1 ภาพ Agent Architecture
ทีม FC Portugal Description

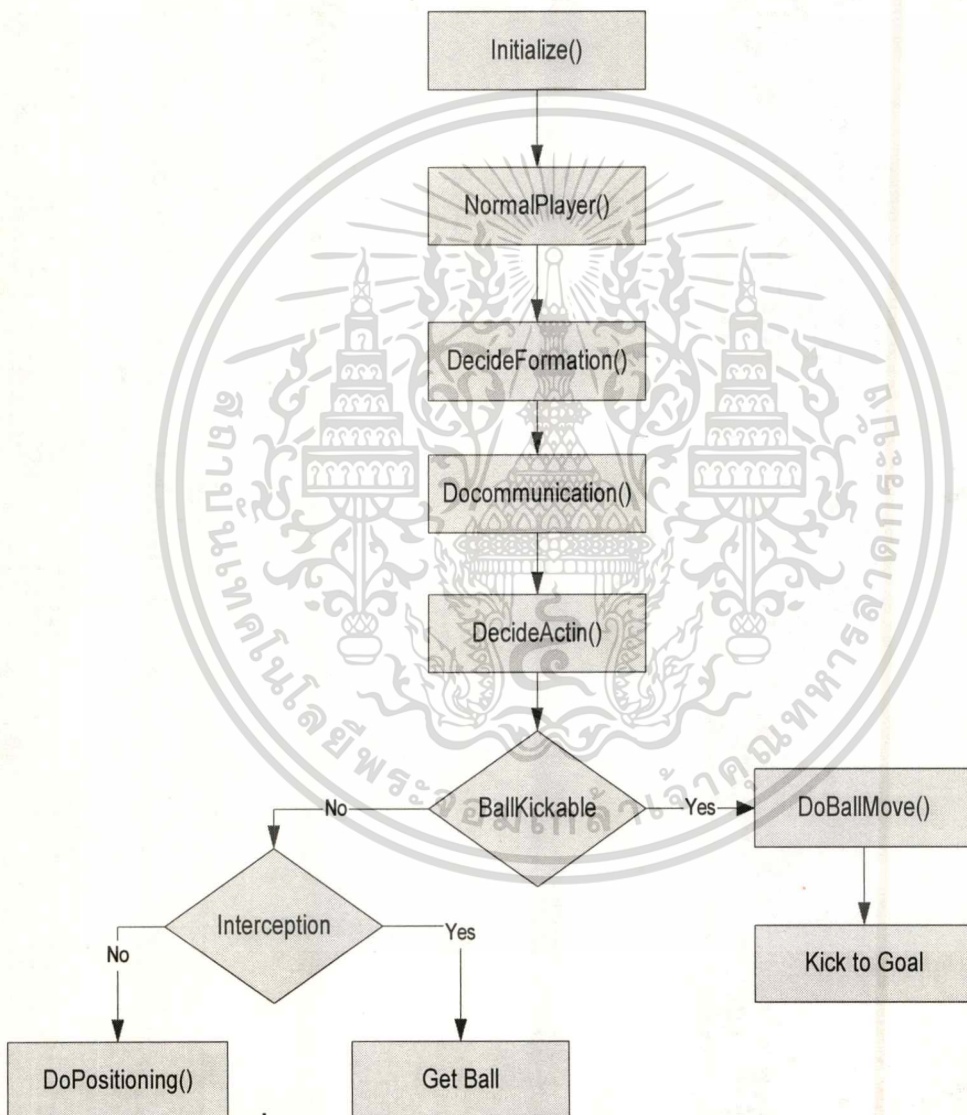
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 Control Flow of the High Level Decision Module
ที่มา FC Portugal Description

จากรูปที่ 4.1 จะแสดงให้เห็นว่า โครงสร้างของ Agent มีองค์ประกอบใดบ้าง และ รูปที่ 4.2 จะแสดงการทำงานเป็น Flow ของขั้นตอนการทำงานและระบบการตัดสินใจของ ทีม FCPortugal โดยอาศัยหลักของ High Level Decision Module

4.2 Process การทำงานของโปรแกรม FCPortugal Agent



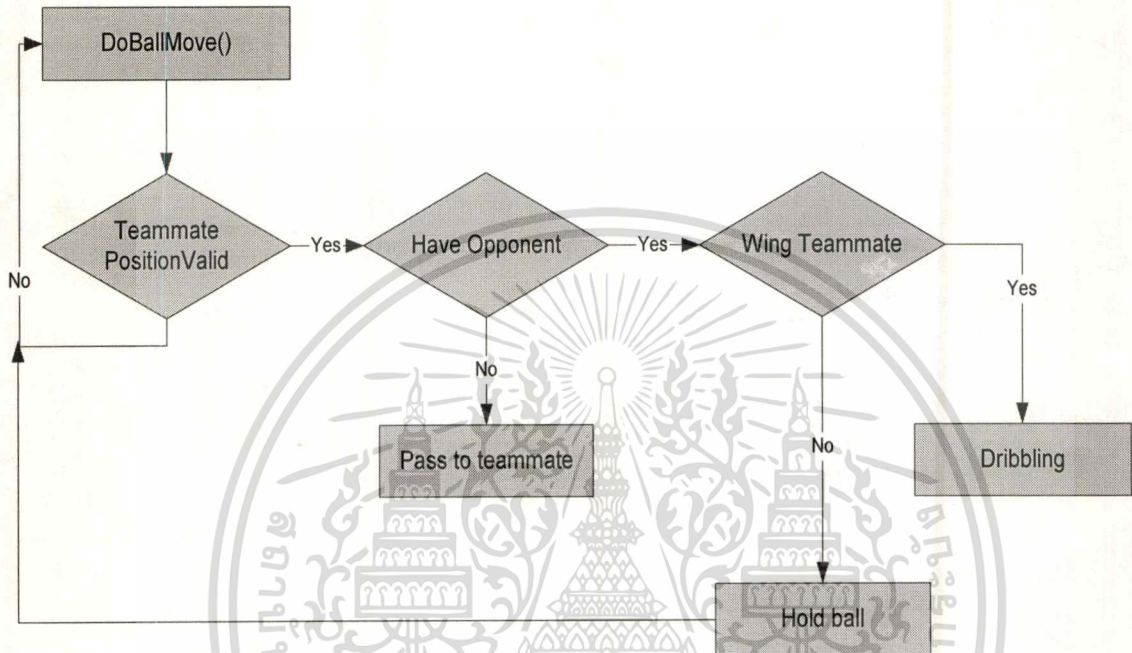
รูปที่ 4.3 การทำงานของ Program FCPortugal Agent

ที่ทีม FC Portugal Description

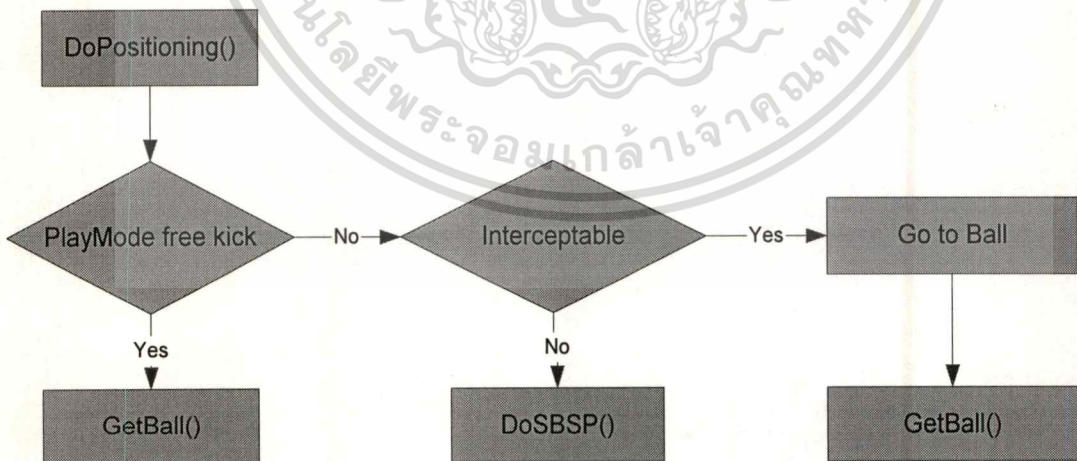
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Flow Chart แสดงการทำงานของโปรแกรมจะแสดงให้เห็นได้ว่าเราจะสามารถดัดแปลงและ

พัฒนา Agent ของ Team ให้ดีขึ้นได้อย่างไรจาก file MemPortugal.c

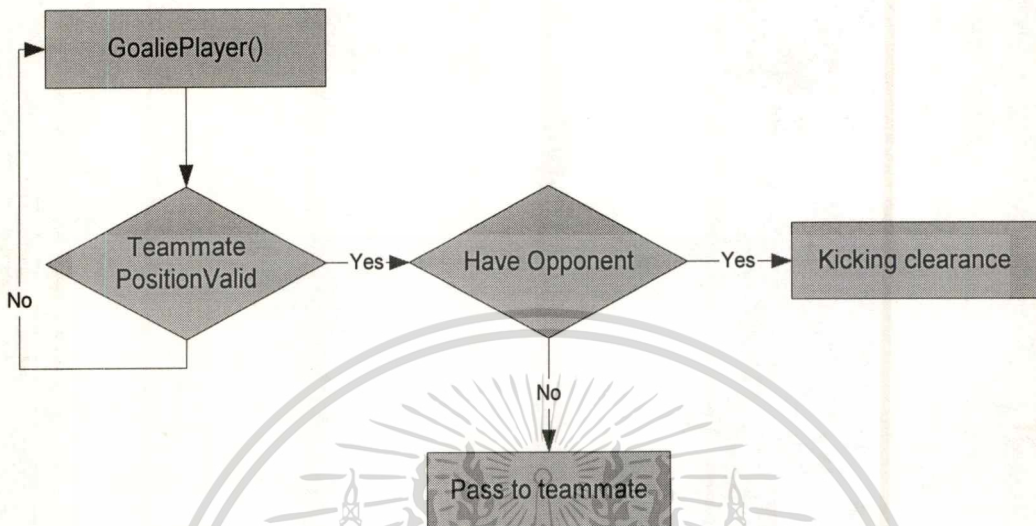


รูปที่ 4.4 การตัดสินใจเมื่อมีบอลของ Team KMITL Agent



รูปที่ 4.5 การจัด Position ของ Team KMITL Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 Process การตัดสินใจของ Goal Team KMITL Agent

4.3 ส่วนประกอบของโปรแกรม

โปรแกรม FC Portugal ประกอบไปด้วย 4 ส่วนใหญ่ ๆ ได้แก่

- OptionInfo : เก็บค่าของ Client Parameter ซึ่งจะถูกเก็บอยู่ใน file ที่มีชื่อว่า MemOption.c
- PlayerInfo : เก็บข้อมูลของ Client เช่น Speed , Stamina ซึ่งจะถูกเก็บอยู่ใน file ที่มีชื่อว่า MemPlayer.c
- PositionInfo : เก็บตำแหน่งของลูกบอล ผู้เล่น ซึ่งจะถูกเก็บอยู่ใน file ที่มีชื่อว่า MemPosition.c
- ActionInfo : เก็บฟังก์ชันการ Intercept ซึ่งจะถูกเก็บอยู่ใน file ที่มีชื่อว่า MemAction.c

4.3.1 รายละเอียดของ file ต่าง ๆ ที่ใช้ภายในโปรแกรม

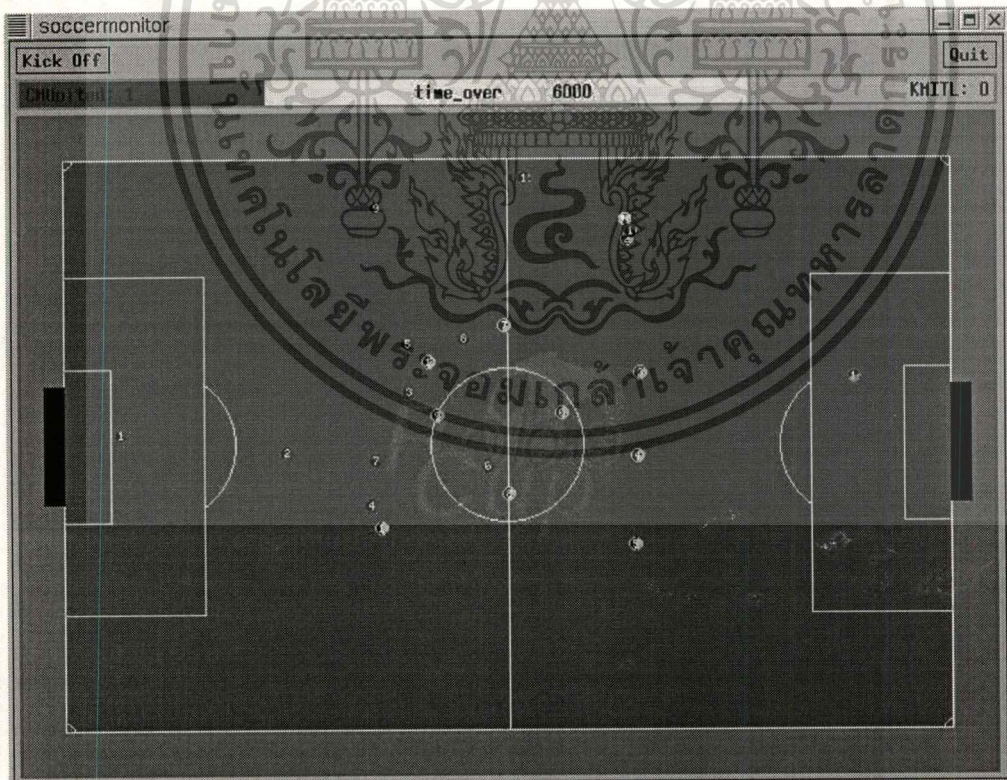
- client.c เป็น file ที่เก็บโปรแกรมหลักของ Agent
- netif.c เป็น file ที่ทำหน้าที่ในการติดต่อกับ Server
- geometry.c เป็น file กำหนดโครงสร้างการคำนวณในลักษณะเรขาคณิต
- utils.c เป็น file ที่สร้าง functions เพื่อใช้ในการทำงานเกี่ยวกับเวลา
- parse.c เป็น file ที่เก็บ messages ที่ออกมาจาก server
- types.h เป็น file ที่กำหนดคสัญลักษณ์ต่าง ๆ ของการเล่นของ Agent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MemOption.c เป็น file ที่กำหนดค่าของ Client Parameter
- MemPlayer.c เป็น file ที่เก็บข้อมูลของ Client เช่น Speed , Stamina เป็นต้น
- MemPosition.c เป็น file ที่เก็บข้อมูลของตำแหน่งของลูกบอล ผู้เล่นฝ่ายเดียวกันและฝ่ายตรงข้าม
- MemAction.c เป็น file ที่เก็บ functions ของ ball interception
- Memory.c เป็น file ที่เก็บ top-level Memory class ของโปรแกรม
- kick.c เป็น file ที่เก็บ functions ของการ kicking the ball
- dribble.c เป็น file ที่เก็บ functions ของการ dribbling the ball
- mybehave.c เป็น file ที่ตัดสินใจการแสดง Action เพื่อส่งให้กับ server
- MemPortugal.c เป็น file ที่เก็บการทำงานหลักของ Agent เพื่อใช้ในการตัดสินใจ

4.4 การพัฒนา KMITL Agent Team

การพัฒนาทีม KMITL ได้นำเอา Function ต่าง ๆ ของ FC Portugal มาใช้ในการพัฒนาเป็นหลัก



รูปที่ 4.7 ภาพการแข่งขันระหว่าง KMITL กับ CMUnited

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การ Compile Client Program

การ Compile Client จะใช้หลักการของการ Compile ภาษา C บน UNIX โดยจะมี MakeFile เป็นตัวบอกถึงการสร้าง File Object เพื่อนำมา link ให้เป็น File Execute เมื่อเราสร้างหรือพัฒนา Agent เสร็จแล้วตัว FCPortugal จะมี Makefile มาให้ด้วยเราสามารถนำมาปรับเปลี่ยนให้ตรงกับ Agent ที่เราทำขึ้นมาได้ โดยเริ่มแรกจะต้องใช้คำสั่งดังนี้

gmake depend เป็นการสั่งให้ตรวจสอบ file library ต่าง ๆ ที่ต้องใช้ในการ compile program

\$ gmake depend

gmake เป็นคำสั่งที่ใช้ในการ Compile และ link object เพื่อสร้าง execute file

\$ gmake

เมื่อทำคำสั่ง gmake ผ่านแล้วจะได้ execute file สำหรับเรียก Agent ให้ทำงานบน Soccer Server ได้

4.6 คำสั่งในการพัฒนาโปรแกรม Agent

ในการพัฒนา Agent นั้นจำเป็นที่จะต้องเข้าใจโครงสร้างของ Function และ Method ที่มีอยู่ภายในโปรแกรมที่ให้มาเป็น Source Code โดยจะบรรจุอยู่ในไฟล์ต่าง ๆ และไฟล์ MemPosition.h นั้นจะเก็บมีคำสั่งที่ให้มีการ Get information เกี่ยวกับ ตำแหน่งของ object ที่อยู่บน Soccer Server โดยจะจัดแบ่งเป็น Ball ,Player และ Marker (Stationary Object เช่น จุดโทษ,ประตู เป็นต้น) คำสั่งเกี่ยวกับ Ball ได้แก่

- BallX() , BallY() คือการให้ค่าของตำแหน่ง Ball ขณะนั้นออกมาแบ่งเป็นแกน X และแกน Y ซึ่งในโปรแกรมนี้จะกำหนด class ของตำแหน่งในแกน X และ แกน Y เป็นชนิด Vector ดังนั้นถ้าเราต้องการจะนำค่าของตำแหน่ง Ball มาใช้จะต้องใช้คำสั่งดังนี้

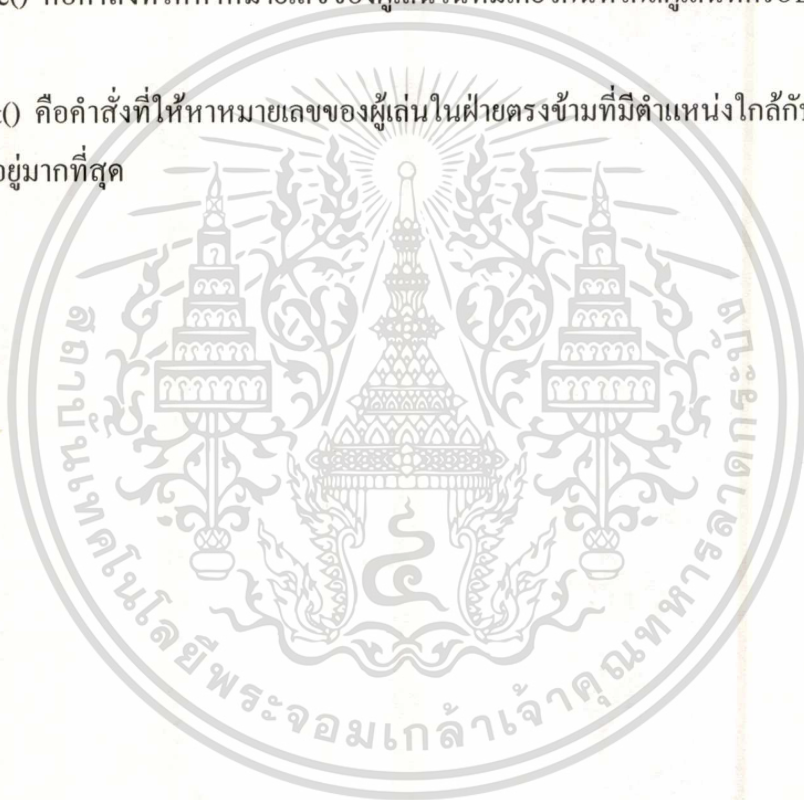
Vector BallPosition ; /ประกาศค่าตัวแปร BallPosition ให้เป็น Vector class

BallPosition.x = BallX(); /ให้นำค่าของตำแหน่ง Ball ในแกน X ไปเก็บไว้ที่ BallPosition.x

BallPosition.y = BallY(); /ให้นำค่าของตำแหน่ง Ball ในแกน Y ไปเก็บไว้ที่ BallPosition.y

- TeammateX(number) ,TeammateY(number) คือการหาตำแหน่งของ Teammate ตามหมายเลขของผู้เล่นในฝ่ายเดียวกับเราที่กำหนดไว้ในค่าของ number (1-11)
- TeammateDistance(number) คือการหาระยะห่างระหว่างตัวผู้เล่นที่ครอบครองบอลขณะนั้นกับตัวของผู้เล่นที่กำหนดไว้ในค่าของ number (1-11)

- OpponentX(number),OpponentY(number) คือ การที่สั่งให้หาตำแหน่งของ ผู้เล่นฝ่ายตรงข้ามตาม หมายเลขที่กำหนดไว้ใน number (1-11)
- MarkerX(markertype),MarkerY(markertype) คือการหาตำแหน่งของ Marker ตามที่ได้กำหนดไว้ในค่า ของ markertype ซึ่งเป็นตำแหน่งต่าง ๆ บนสนามและเก็บไว้ในไฟล์ type.h โดยมีการอ้างอิงมาจาก Soccer Monitor ที่สร้างสนามแข่งขันจาก Soccer Server
- ClosestTeammate() คือคำสั่งที่ให้หาหมายเลขของผู้เล่นในทีมเดียวกันที่ใกล้ผู้เล่นที่ครอบครองบอล มากที่สุด
- ClosestOpponent() คือคำสั่งที่ให้หาหมายเลขของผู้เล่นในฝ่ายตรงข้ามที่มีตำแหน่งใกล้กับผู้เล่นฝ่ายที่ ครอบครองบอลอยู่มากที่สุด



บทที่ 5

บทสรุป

จากการศึกษาการวิเคราะห์และออกแบบระบบ Multi Agent System ที่นำมาใช้ในการสร้าง Agent สำหรับ Robo Cup ผ่านทางการทำงานของ Soccer Server โดยใช้เทคโนโลยีเชิงวัตถุที่มีขั้นตอนการพัฒนาหลายขั้นตอน ตั้งแต่การกำหนดแบบแผนในการแข่งขัน การออกแบบกฎการแข่งขัน และการ Implement Agent โดยใช้ ภาษา C++ ภายใต้ Environment ของ LINUX สามารถสรุปผลการศึกษาได้ดังต่อไปนี้

5.1 ผลการพัฒนา Agent

Agent ของทีม KMITL จะมีลักษณะในการเล่นเกมที่ในรูปแบบ 433 เป็นหลัก โดยกำหนดการวางตำแหน่งของทีมด้วย SBSP ให้มีการยืนรักษาตำแหน่งและเคลื่อนที่ตาม Formation ในระหว่างที่ Agent นั้นไม่มีลูกบอลครอบครอง และใช้หลักการตัดสินใจจากการครอบครองบอลซึ่งได้แสดงในรูปที่ 4.3 จากการลองทดสอบการแข่งขันกับทีม FCPAgent และทีม CMUnited ผลออกมาดังนี้



รูปที่ 5.1 การแข่งขันระหว่าง ทีม KMITL กับทีม FCPortugal

KMITL 0 : 1 CMUnited

KMITL 0 : 1 CMUnited

KMITL 4 : 4 FCPlayer

KMITL 4 : 5 FCPlayer

KMITL 4 : 3 FCPlayer

KMITL 5 : 3 FCPlayer

5.2 ประโยชน์ที่ได้รับ

ระบบ Multi Agent System ระบบที่ค่อนข้างมีความซับซ้อนและมีความหลากหลาย เนื่องจากมีความเกี่ยวข้องกับด้าน Computer Artificial Intelligent จึงต้องมีความเข้าใจในการทำงานของ Multi Agent System เพื่อให้การพัฒนาเป็นไปอย่างรวดเร็ว ซึ่งการออกแบบ Agent เพื่อนำมาใช้ในการแข่งขัน Robo Cup นั้นก็ได้นำเอาหลักต่าง ๆ ของ Multi Agent System มาใช้ จากการศึกษา Agent ของ CM United และ FC Portugal ได้เรียนรู้ถึงความสัมพันธ์และข้อดีข้อเสียต่าง ๆ แล้วได้นำข้อมูลที่ได้เหล่านั้นมาใช้ในการปรับปรุงและการพัฒนา Agent ของโครงการนี้ นอกจากนี้ยังได้เรียนรู้ Soccer Server และ ระบบปฏิบัติการ LINUX สำหรับใช้ในการพัฒนาด้วย

5.3 ข้อเสนอแนะ

การพัฒนา Agent มาใช้ในการแข่งขันนั้นมีความยุ่งยากในการที่จะทำความเข้าใจทั้งหมด ดังนั้นจึงควรที่จะศึกษาจาก Agent team ที่มีให้ Download ตามอินเทอร์เน็ต ที่มีเป็น Low Level Module จากนั้นจึงนำ Code มาพัฒนาส่วนที่เป็น High Level Module ของแต่ละ Team ให้มีความยืดหยุ่นในการแข่งขันและสามารถที่จะเลือกการตัดสินใจที่มีประสิทธิภาพต่อไป ซึ่งในส่วนของ Algorithm ที่นำมาใช้กับ Agent นั้นเราสามารถที่จะนำเอาหลักการทำงานในลักษณะของ Artificial Intelligent (AI) มาใช้ช่วยในการพัฒนาได้ เช่น Reinforcement Learning , Decision Tree เป็นต้น

บรรณานุกรม

FC Portugal Homepage. 2000.[Online], Available : <http://www.ieeta.pt/robocup>

Luis Paulo Reis and Nuno Lau. 2000, **FC Portugal Team Description : RoboCup 2000 Simulation LeagueChampion** ,Universitiy of Porto , University of Aveiro, Portugal

Perter Stone .1998, **Layered Learning in Multi-Agent System**, Ph.D. Thesis , School of Computer Science Mellon University , Pittsburgh

Peter's Home Page. 2000.[Online]. Available : <http://www.research.att.com/~pstone>

Robocup Official Site. 2000.[Online]. Available : <http://www.robocup.org>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นายสุรเชษฐ เดชระนังทอง
สถานที่เกิด	จังหวัดจันทบุรี
วุฒิการศึกษา	วิทยาศาสตรบัณฑิต คณะวิทยาศาสตร์ สาขาสถิติประยุกต์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง
ตำแหน่งหน้าที่	Network Engineer
สถานที่ทำงาน	บริษัท The Communication Solution (TCS)

