Soccer Strategy

Sascha Lange, Christian Müller, Stefan Welker

Brainstormers Tribots

Neuroinformatics Group
University of Osnabrück
Albrechstr. 28
49069 Osnabrück

Tel: +49 541 969-2390
Fax: +49 541 969-2246

http://www.tribots.uos.de
Email: tribots@informatik.uni-osnabrueck.de
Motivation

- Cooperation in the Tribots up to RC 2004:
  - treated more than an annoying necessity than a chance
  - no general concept but collection of N individual solutions
Some deficiencies:

- WLan break-down: No robot approached the ball
- Implementation of coordinated plays nearly impossible, coordination had to be spread to nearly everywhere
- no dynamic help (in case one robot was beaten)
Motivation

- Very primitive, „static“ team strategy
- each robot formed an individual „line of defense“
- each robot became the single striker in case it gained possession of the ball
- robots without ball possession did not make a difference between attacking and defending
Motivation

- New goal (since 2005): Implement a complex team defense strategy in the Tribots
  - keep a complex formation while moving
  - coordinated action in sub-teams
    - double team
    - passing (in offense)
  - dynamically providing help, dynamic role changes
Example: 2-2-1 Pressure Defense

CONNECTICUT’S 2-2-1 PRESSURE DEFENSE
Jim Calhoun
NCAA Champion Coach

Since Jim Calhoun’s arrival at the University of Connecticut, the 2-2-1 zone press defense has been used with remarkable effectiveness. Coach Calhoun’s 2-2-1 plays an integral role in a Connecticut defense that has been the highest ranked defense in the country. The defense can be explained as follows:

1. Don’t reinvent the wheel: use defensive team strategies from “real” sports for the Tribots
2. Transfer concepts and methods coaches would use for implementing such a team strategy in a team of human players
Example: 2-2-1 Pressure Defense

- Adapted the original zone pressure to the MidSize League
- These diagrams depict the basis of the plan, we wanted to implement
Playertypes in the Tribots

- Playertypes hook in the architecture as the „decision making“ module

- can be (manually) changed during runtime (all types are available to each robot, abstract factory) but do not change automatically / within a game

- Playertypes used during a soccer game:
  - fieldplayer (all robots despite the goalie)
  - goalie

- there is no communication / cooperation between players of different types

➡ in this talk we only consider the fieldplayers
Distinction made between

„Explicit“ Cooperation and

„Implicit“ Cooperation / Coordination
Cooperation in the Tribots

- "Explicit" Cooperation
  - involves communication; e.g. commanding, contracting, bidding, etc.
  - decisions made during game
  - AI-techniques, done by the robots
  - latency, bandwidth usage, hard to predict, take care of ambiguities / oscillations
Cooperation in the Tribots

- „Implicit“ Cooperation / Coordination
- coordinating player’s behavior during implementation phase, simply by knowing and adapting all other players’ strategies
- decisions made before game
- human, not done by the robots
- no latency, no bandwidth usage, oscillation may occur, not „adaptive“
- „Mirroring“ (calculating the other player’s strategy to adapt own decisions to their strategy)
Cooperation in the Tribots

- Distinction made between
  - "Explicit" Cooperation and
  - "Implicit" Cooperation / Coordination

→ we try to avoid explicit cooperation wherever possible
Roles in the Tribots

- What is a „role“?
  - collection of responsibilities (motivated by real sports)
  - may be switched dynamically
  - existing roles defined before the game
  - we use roles mainly as tool for implicit coordination
  - optionally map player-roles to roles of specific tactics
    - field-player-roles to zone-pressure-roles (presently id)
    - field-player-roles to standard-situation-roles
  → thus, we have different sets of roles in the strategy
Roles in the Tribots

- ballL, ballR:
  - closest to the ball (left side, right side)
  - should try to get possession of the ball
  - "strikers", first-line-defenders
- left, right
  - further away from the ball
  - should block shots to the goal
  - "defenders", second-line-defenders
- safety
  - stays at the own penalty area
  - last-line of defense
Roles in the Tribots

- How do we realize roles?
  - assigned role is just an additional information in the „state“
  - roles are defined for each playertype, e.g. goaly does not care about roles of fieldplayers
  - no separate code / behavior stacks for different roles
- Influence of roles on the strategy:
  - Modification of the output of a behavior. Example: protect goal.
  - Modification of the activation of a behavior. Example: block the sideline.
Roles in the Tribots

BProtectGoal:

- most simple modification of the behavior
- both roles (left&right) use exactly the same behavior
- only some reference point is changed

```cpp
DriveVector getCmd(const Time& t) throw(TribotsException)
{
... if (WBOARD->getZonePressureRole() == "left") {
    protectPos = Vec(-0.85*field.goal_width/2.,-field.field_length/2.);
} else { // "right"
    protectPos = Vec(+0.85*field.goal_width/2.,-field.field_length/2.);
}
... return BProtectGoal::getCmd(t);
}
```
Roles in the Tribots

- How do we realize roles?
  - assigned role is just an additional information in the "state"

- roles are defined for each playertype individually, e.g. "goaly" does not care about roles of "fieldplayers"

- no separate code / option stacks for different roles

- Influence of roles on the strategy:
  - Modification of the output of a behavior. Example: pong.

  - Modifikation of the activation of a behavior. Example: passreceiver.
Roles in the Tribots

**BSupportDoubleTeamSideline:**

- most simple modification of the activation
- both roles (left&right) use exactly the same behavior
- only the activation depends on the position of the ball and the own role

```cpp
bool checkInvocationCondition(const Time& t) throw() {
    ...
    return
        ((WBOARD->getZonePressureRole() == "left" &&
           ballLocation.pos.x < -500) ||
         (WBOARD->getZonePressureRole() == "right" &&
           ballLocation.pos.x > +500));
}
```
FieldPlayer

- BGameStopped
- BLeaveGoal
- BPreOwnIndirectStandardSituation
- BOwnPenalty
  - BOwnPenaltyGotoCenter
  - BOwnPenaltyGotoSpot
  - BOwnPenaltyDecide
  - BOwnPenaltyGotoKickPos
  - BOwnPenaltyWaitBeforeKick
  - BOwnPenaltyPreApproach
  - BOwnPenaltyApproach
  - BOwnPenaltyKick
- BStuckStandardSituationFP07
- BVolleyApproachAfterOwnSetPlay
- BPostOpponentStandardSituation
- BPreOpponentStandardSituation
- BSTayInsideArea
- BCounterAttack
- BallOwnerStack
  - BShootEmergency
  - BFarShoot
  - BDraufhalten
  - BShoot
  - BBefreiungsschlag
  - BStuckOwnsBallConditioned
  - BPasinFrontOfGoal
  - BPassSpontaneously
  - BStuckDistanceShooter
  - BTouchBallAfterStandard
  - BallPassingReceiver
  - BOpposeBall
  - BInterceptBallStatic
  - BEigenMove
  - BRetreatDribble
  - BShakeOffDefender
  - BWingAttack
  - BBoostBallToGoal
  - BDribbleBallStraightToGoalEvadeSidewards
  - BDribbleBallToGoal
  - BComplexApproachBallFreePlay
  - BAvoidGoalieArea
  - BApproachBallAfterNonexecutedStandard
  - BApproachBallDirectlyAfterStandard
  - BInterceptBallRL
  - BInterceptBall
  - BSupportLongPassConditioned
  - BSupportNearBallConditioned
  - ZonePressure
  - BDefendBall
  - BDoubleTeam
  - BSupportDoubleTeamSideline
  - BProtectGoal
  - BSupportDoubleTeamMiddle
  - BSafety
  - BPatrolFP07
  - BEmergencyStop
  - BFP07Update
Options in Fieldplayer

BGameStopped
BLeaveGoal
BPreOwnIndirectStandardSituation
BOwnPenalty
BStuckStandardSituationFP07
BPostOpponentStandardSituation
BPreOpponentStandardSituation
BStayInsideArea
BCounterAttack

BallOwnerStack

BSupportLongPassConditioned
BSupportNearBallConditioned

ZonePressure

BPatrolFP07
BEmergencyStop
BFP07Update

Options in BallOwnerStack

BShootEmergency
BDraufhalten
BShoot
BBefreiungsschlag
BPassInFrontOfGoal
BPassSpontaneously
BTouchBallAfterStandard
BallPassingReceiver
BEigenMove
BRetreatDribble
BShakeOffDefender
BWingAttack
BBoostBallToGoal
BDribbleBallStraightToGoalEvadeSidewards
BDribbleBallToGoal
BComplexApproachBallFreePlay

Options in ZonePressure

BDefendBallConditioned
BDoubleTeamConditioned
BSupportDoubleTeamSidelineConditioned
BProtectGoalConditioned
BSupportDoubleTeamMiddleConditioned
BSafetyConditioned
= affected by roles

Options in Fieldplayer

- BGameStopped
- BLeaveGoal
- BPreOwnIndirectStandardSituation
- BOwnPenalty
- BStuckStandardSituationFP07
- BPostOpponentStandardSituation
- BPreOpponentStandardSituation
- BStayInsideArea
- BCounterAttack

Options in BallOwnerStack

- BShootEmergency
- BDraufhalten
- BShoot
- BBefreiungsschlag
- BPassInFrontOfGoal
- BPassSpontaniously
- BTouchBallAfterStandard
- BallPassingReceiver
- BEigenMove
- BRetreatDribble
- BShakeOffDefender
- BWingAttack
- BBoostBallToGoal
- BDribbleBallStraightToGoalEvadeSideways
- BDribbleBallToGoal
- BComplexApproachBallFreePlay

Options in ZonePressure

- BDefendBallConditioned
- BDoubleTeamConditioned
- BSupportDoubleTeamSidelineConditioned
- BProtectGoalConditioned
- BSupportDoubleTeamMiddleConditioned
- BSafetyConditioned
Roles are heavily used in the fieldplayer stack
- behaviors are fine-tuned on each other
- exchangeability is restricted

No roles IN the ball-owner-stack
- when the ball-stack is activated, independent from its role, every player should get the ball, advance it and (ideally) score a goal
- easy to replace / exchange submodules
- most students work on this part (more fun)
**Example: 2-2-1 Pressure Defense**

- Most behaviors are activated depending on the position of the ball.
- Decision boundaries on the field.

Diagram showing decision boundaries and behaviors such as `BProtectGoal`, `BApproachBall`, `BBlockGoal`, `BBlockMiddle`, and `BBlockSideline`.
Some ‘Tricks‘ we Use

- Carefully **overlap / non-overlap** areas of responsibility / activation

- e.g. in the middle of the field overlapping areas of responsibility -> prefer two robots blocking the goal over none blocking the goal

- always use **hysteresis** at „decision-boundaries“ to prevent oscillating

- always **prefer** the „safe“, „game destroying“ **alternative** over the more „beautiful“ but risky alternative

- e.g. risk a „huddle“ of several robots in order to prevent giving the opponent a chance to advance
Keeping the Formation

- Demonstration in Simulator
Teamcontrol distributes the roles

- Some robots may have preferences
- Assigns necessary roles to active robots (constraint satisfaction)
- Reassignment during own standards (according to preferences)
\textbf{\textit{Static} role assignment}

- TC has schemes for less than five players
- Happens often in games
- No degradation in the quality of the strategy
- Roles take over responsibilities of missing roles

E.g. in case of a complete break-down of the communication (only one player), all players will take over „responsibilities of all roles“, thus, all robots will run amok and go to the ball, wherever it is on the field.
Missing players
Explicit Cooperation

- Dynamic role changes
- Defense rotations
- Dynamic Subteams / Small „Set Plays“
  - Passing / Volley
  - Defensive Double Team
- Determination of the „Ball Owner“
- Idea: „Dynamic Chain of Command“ (Dynamic Hierarchy)
Dynamic Chain of Command

- Distributed decision making
- latency / drop in responsiveness, conflicts
- hard to program, hard to change later, hard to keep track!!! ;-)  

⇒ Thus, keep all the decision making in one place
Dynamic Chain of Command

- Simple idea:
  - One master doing all the decision making
  - commanding the slaves
  - Don’t shout plans, shout signal words (sport!)
  - Corresponding action has been preplanned
  - again, implicit coordination but activated through communication
  - in sports: preplanning and signal words are job of the coach
  - „Commander in chief“ (master) may change dynamically, depending on situation
  - It’s possible to have even several commanders at the same time for different aspects of the cooperative strategy
Example: Pass

- BCounterAttack
- BWingAttack
Example: Pass
Example: Pass
individual robot

the robot possessing the ball is the only robot to demand a dynamic change of roles, e.g. if he is a defender and wants to become an attacker
to initiate a coordinated pass play by sending the receivers to their designated receiving positions.
generally, the ball owner is the most important robot for us
individual robot

the robot in the first line of defense on who’s side the ball becomes the „defensive leader“ and is the only robot to

decide, whether the first line of defense has been overplayed along the sideline or through the gap of the DoubleTeam and thus demanding a defensive-rotation (like in real sports!!!)

commanding a teammate to a double-team action
central entity, using the fused world model

- teamcontrol, or placed on one robot (e.g. lowest number)

- commands other robots to not approach the ball (by simply switching off the whole ball stack), if one robot is already close to it and in a good position
Problem: Fused world model on robots isn’t synchronized and ambiguous.

Therefore, we need the most distinguishing criteria e.g.

- ball possession
- ball in area of responsibility (attached to role)
Nevertheless ambiguities may occur

- ball on decision boundary, noisy perceptions
- intentionally overlapping criteria

Our solution (in praxis we talk about a few ms):

- Make use of client / server architecture
- Who shouts first wins
  (discrete steps in teamcontrol)
- If two shout at the same time, the one with the lower shirt number wins
Implementation

Whiteboard:
- plain text, no restrictions
- signal words (if I implement a new play, I invent a new signal)

Activation: In most cases, there is exactly one behavior that asks for activation (IC=true) when hearing one specific signal

Decision making:
- Central hook in Fieldplayer (e.g. used for defensive rotations)
- Cycle callback of corresponding behavior (e.g. in Pass-behavior for the pass-play)
- Cycle callback of activated behavior (remote decision, but encapsulated locally in only one behavior in the source code)
Example: 2-2-1 Pressure Defense

- First line defender on side of ball is responsible for the decision to rotate
- Team reacts with a rotation of a second line defender towards the ball
Example: 2-2-1 Pressure Defense

- first line defender on side of ball is responsible for the decision to double
- other first line defender reacts with activating double-team behavior
Example: 2-2-1 Pressure Defense

In the 2-2-1 Pressure Defense, the ball is typically held in the middle. The defender, X1, will play the ball while X2, X3, and X4 will cover the wings. The deep alleyway, X3, will act as a safety to prevent the ball from getting out of control. The deep alleyway will try to anticipate the ball's movement and block any passing attempts.
Example: 2-2-1 Pressure Defense

- ball possession is communicated constantly in every cycle
- teammates switch to attack mode and activate corresponding behaviors
Last idea: „Mirroring“

- Wanted to build a „Mauer“
- no gaps between robots
- visual avoidance strategy leads to oscillations

➡ „Mirroring“

➡ calculate the strategy and desired positions of all the other roles and slightly adapt own desired position

➡ use chain of command to determine „ordering of avoidance“
Special Situation: Opp Free Kicks
Special Situation: Opp Free Kicks
Special Situation: Opp Free Kicks
That's it!

- implicit / explicit coordination
- static and dynamic role changes
- dynamic chain of command
- mirroring

- no secrets left
- complete source code will be made available in december
### Ball Distribution

#### RC08 Semi-Final vs. Eindhoven

<table>
<thead>
<tr>
<th></th>
<th>5.7</th>
<th>7.4</th>
<th>4.3</th>
<th>4.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3883</td>
<td>5019</td>
<td>2950</td>
<td>2921</td>
</tr>
<tr>
<td>7.1</td>
<td>4862</td>
<td>7.0</td>
<td>3361</td>
<td>4967</td>
</tr>
<tr>
<td>4.7</td>
<td>3877</td>
<td>3244</td>
<td>2381</td>
<td>1833</td>
</tr>
<tr>
<td>6.1</td>
<td>4153</td>
<td>4841</td>
<td>3114</td>
<td>1635</td>
</tr>
<tr>
<td>2.7</td>
<td>1889</td>
<td>1718</td>
<td>297</td>
<td>617</td>
</tr>
<tr>
<td>3.3</td>
<td>2283</td>
<td>463</td>
<td>687</td>
<td>1899</td>
</tr>
</tbody>
</table>

Initial play:
The ball is not on 19.37. Tight press after the opponent's half ball, to the strong force ball back to the middle of the ball. Second-line defense must then pin the middle and encourage the defender to follow the ball. X4 marks the depth of the ball, X3 protects the ball, X2 follows the ball. It is important where the trap is.

#### RC08 Small-Final vs. COPS

<table>
<thead>
<tr>
<th></th>
<th>4.1</th>
<th>2.2</th>
<th>0.5</th>
<th>0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3095</td>
<td>1651</td>
<td>381</td>
<td>55</td>
</tr>
<tr>
<td>3.7</td>
<td>2841</td>
<td>2506</td>
<td>1656</td>
<td>1770</td>
</tr>
<tr>
<td>2.0</td>
<td>1509</td>
<td>2844</td>
<td>2736</td>
<td>7274</td>
</tr>
<tr>
<td>5.9</td>
<td>4484</td>
<td>6869</td>
<td>7026</td>
<td>5573</td>
</tr>
<tr>
<td>4.0</td>
<td>3016</td>
<td>3882</td>
<td>4754</td>
<td>3698</td>
</tr>
<tr>
<td>2.3</td>
<td>1776</td>
<td>925</td>
<td>1269</td>
<td>3334</td>
</tr>
</tbody>
</table>

We will make the trap after the opponent's half ball, to the strong force ball back to the middle of the ball. Second-line defense must then pin the middle and encourage the defender to follow the ball. X4 marks the depth of the ball, X3 protects the ball, X2 follows the ball. It is important where the trap is.
Robot Distribution

Small Final vs. COPS

6. Challenge every shot. When the ball is advanced against our press and a shot is taken, it must be challenged aggressively.

7. Rebound. We must secure the rebound and limit second-opportunity scores.

---

Initial Setup, Slides, and Rotations

The ball is not deflected off the board. X1 and X2 pinch to the middle level of the ball as it is second-line defender of hedge and recover and encourage a look. X4 and player must slide deep to the middle. X5 passes to the deep middle, what we call the "short middle" position. X4 moves to the deep side of the court, stretching the defense and allowing our deep player to attack the basket for protection in what we call the "deep position.

This is a good example of our..."

---

ballL

ballR

---

safety

---

4% 8% 12% 16% >20%

---

Figure 19.38
Semi Final vs. Eindhoven

Initial Setup, Slides, and Rotations

The ball is not denied on the inbounds (see Figure 19.37). Tight pressure is immediately applied on the ball after it is inbounds on the half out. The pincher of the middle has a level of the ball as it is at second-line defender, X2 of hedge and recovery at and encouraging a lob pass defender, X1, must play a rear. The deep man-side to defend the deep to a front-court offensive back to the basket. It is to go to the middle is a fairly out of the middle for ball must go around our.

We do not discourage the because we do not believe that will. After our first trap we will try to play a unique feature that is entire significantly to the trap once before a trap a number of times taking off the press. Another point we emphasize is that we do not believe the long slide is very difficult to throw by ball handler.

On a ball reversal we will employ the fundamental "bump" principle and make the appropriate rotations (see Figure 19.38). As the ball is reversed, the entire zone made in the alley as long as it is a trap. We tell our players that "we want a good trap, not necessarily an early one." By a "good trap" we mean an aggressive, solidly executed trap in which our defenders have their hands up, put pressure on the ball, and keep pressure on it. We will try to have him to pass—we make the trap a pass one of the possible. Over one we tell our. Another area is that broken. The ship work very important. In seekball or court. We feel the style coach Cali through no safety.
Eindhoven

Brainstormers Tribots, Neuroinformatics Group, University of Osnabrück, 2008
www.tribos.uos.de