EVALUATION OF FANTASY PLAYERS

Inventors: Scott B. Ware, Santa Clara, CA (US); Brian A. Webb, Sunnyvale, CA (US)

Correspondence Address:
YAHOO C/O MOFO PALO ALTO
755 PAGE MILL ROAD
PALO ALTO, CA 94304

Assignee: Yahoo! Inc., Sunnyvale, CA (US)

Appl. No.: 11/606,717

Filed: Nov. 29, 2006

Publication Classification

Int. Cl. A63F 9/24 (2006.01)

The evaluation of players in the context of a fantasy sports league is disclosed. In order to evaluate whether a fantasy team owner should add one player to the starting lineup of the team and drop another, player stat predictions are first used to predict end-of-season stats for the current starting lineup of a team. Historical and current league stat/point functions may then be used to determine how many points the team is expected to earn during the course of a full season based on the predicted end-of-season stats. This same methodology is then applied to a proposed revised lineup to determine how many points the revised team is expected to earn during the course of a full season. The difference between the projected point totals for the current team and the revised team are an indication of the value of this specific transaction to this specific team.
<table>
<thead>
<tr>
<th></th>
<th>HR</th>
<th>H</th>
<th>SB</th>
<th>W</th>
<th>K</th>
<th>Total Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>2</td>
<td>3</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>5</td>
</tr>
<tr>
<td>Team B</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>Team C</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

**FIG. 1**

<table>
<thead>
<tr>
<th></th>
<th>HR</th>
<th>H</th>
<th>SB</th>
<th>W</th>
<th>K</th>
<th>Total Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>100</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>Team B</td>
<td>110</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team C</td>
<td>90</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Points**
Revised Starting Lineup 206

Adjust Lineup

Current Starting Lineup 204

Other Players 202

Generic Player Rankings 200

FIG. 2
Revised Starting Lineup

Historical League Stats → Points Data

Current League Stats → Points Data

Stats-to-Points Conversion

Player Stat Predictions

Other Players + Current Starting Lineup

Adjust Lineup

Revised Starting Lineup

FIG. 3
FIG. 4
EVALUATION OF FANTASY PLAYERS

BACKGROUND OF THE INVENTION

Fantasy sports leagues have become a hugely popular form of recreation and competition for sports enthusiasts. In general, fantasy sports leagues contain a number of fantasy teams, each team managed by a fantasy team owner. The role of an owner is to manage a virtual team of real-life professional sports players and decide which players to sit, start, drop, add, and trade with other teams. The goal is to have the starting players on the fantasy team accumulate the most positive statistics (“stats”) in their real-life games. These accumulated stats are directly converted to fantasy points using a rotisserie-style scoring algorithm. The team with the most points at the end of the season is declared the winner of the league.

Rotisserie style scoring is accomplished by ranking all teams from 1 to N, where N is the number of teams in league, for each stat category. For any given stat category (e.g. home runs (HRs), hits (H) and stolen bases (SB) for non-pitchers, or strikeouts (K) and wins (W) for pitchers in a fantasy baseball league), the first place team will get N points. The second place team will get N-1 points. This continues for each team in the league and is repeated for each stat category. The team points in most scoring systems are simply the sum of the team points in all stat categories combined.

FIG. 1 is an exemplar chart illustrating the rotisserie style scoring system in the context of a fantasy baseball league. In FIG. 1, lower table 100 shows the actual combined statistics of players on each team in a three team league at a particular point in the season. As the lower table illustrates, team B leads in HR, followed by A and C, but team A leads in SB, followed by teams B and C. As upper table 102 illustrates, for the HR stat, the HR totals translate into 3 points for team B, 2 points for team A, and 1 point for team C. Similarly, the SB totals translate into 5 points for team A, 2 points for team B, and 1 point for team C. Using just these two stat categories as an example, teams A and B are tied with 5 points each, followed by team C with 2 points.

To successfully manage a fantasy team, an owner must constantly examine different players and evaluate how they can contribute to the stats earned by the fantasy team. There are many different ways that owners try to value players. An owner can read news and notes, view player stats, and consume expert advice. There are also many types of player rankings that attempt to value players. Player rankings can be compiled programmatically by looking at historical stats and/or running simulations, or manually crafted by domain experts. These rankings are either general for all fantasy leagues, or customizable for a specific fantasy league, because each league may have different rules and a different scoring methodology or configuration. Additionally, these rankings usually have some subjective way to evaluate player performance (e.g. stats earned) and create a single ranking order. A typical process for making managerial decisions might be: (1) examine the rankings of players that are available but not currently on the team, (2) examine the rankings for players currently on the team and in the starting lineup, and (3) change the starting lineup to optimize the rankings of players in the lineup.

FIG. 2 is a flow diagram that attempts to illustrate the selection of players for a team in a conventional fantasy league as described above. In FIG. 2, generic player rankings 200 may be published by so-called “experts” in the sport, based on any number of methodologies, in an attempt to give each player a value. The basis for these rankings may include individual historical performance, and in particular, performance against upcoming opponents, performance adjusted for seasonal variations, performance in home or away locations, performance in day or night games, etc. A fantasy team owner has a pool of players 202 available to be added to his or her team’s starting lineup, including “free agents” (players not selected by any team), players on other teams that may be acquired via a trade, and players on the team but not in the starting lineup. The task of any owner is to choose a starting lineup prior to each game. Players can be added to or dropped from the team at various intervals (e.g. before the season during a “draft,” or on a weekly or daily basis). Once on the team, players can be added to or dropped from the starting lineup before the next actual game. The selection of players (via drafting, adding, dropping, or trading) may be based on the generic player rankings or any methodology chosen by the team owner. The starting lineup is represented by the “players in my lineup” block 204, and this lineup may be adjusted (see arrow) to create a revised lineup (see revised lineup block 206) by considering the generic player rankings block. The goal of the owner is to optimize the revised lineup based on the generic player rankings, and compile a team whose individual and overall ranking is the greatest in the hope that the highest ranked players will yield the most team points at the end of the season.

One problem with this approach is that fielding a team of players with optimal player rankings is not always the best way to try to win a rotisserie style fantasy league. The generic player rankings only provide a ranking of players based on the moment in time that the rankings are published, and rank players either according to position or according to selected stat categories. These rankings do not provide the owner or manager with rankings based on the ultimate criter- ia, points, and do not provide an assessment of how a particular player being considered for the team will affect the team’s overall point totals. For example, suppose an owner’s team is leading in HRs and expects to lead the league in HRs by a wide margin, but is close to other teams in SBs. When deciding whether to add player A and drop player B, the owner needs to understand that if adding player A causes HR totals to drop somewhat, but the team will still lead the league in HRs, no points are lost. However, if adding player A causes the team to add enough SBs to leapfrog other teams, points are gained. Thus, overall it makes sense to add player A and drop player B. In other words, because fantasy teams are ranked according to points only, it would be preferable for a fantasy owner to get an idea of how many points a player is expected to contribute to the team over the full season.

Therefore, it would be advantageous to provide a fantasy team owner with a tool for evaluating a player being contemplated for addition to a fantasy team in accordance with how that player is expected to affect the team’s point total at the end of a season, which is the ultimate fantasy league measure of team success.

BRIEF SUMMARY OF THE INVENTION

Embodiments of the invention concern the evaluation of players in the context of a fantasy sports league. A player value, personalized for a specific fantasy team, is generated for a particular player being contemplated for addition to the starting lineup of that team. This value informs a team owner how that player is expected to affect the fantasy points
earned by the team at the end of a season, and allows the owner to make managerial decisions optimized for earning the most fantasy points and winning the fantasy league.

For example, at any point in a current season, in order to evaluate whether a fantasy team owner should add one player to the starting lineup of the team and drop another, player stat predictions are first used to predict end-of-season stats for the current starting lineup of a team. Historical and/or current league stat/point functions may then be used to determine how many points the team is expected to earn during the course of a full season based on the predicted end-of-season stats. The historical and current league stats may be utilized in accordance with a blending function, which gives more weight to historical league stats early in the current season, and more weight to current league stats later in the season.

Next, projected stats for the player under consideration for being dropped may be subtracted from the current team predicted end-of-season stats, and the projected stats of the player under consideration for being added to the current team may be added to the current team predicted end-of-season stats. These revised team stat totals may once again be applied to the historical and current league stat/point functions to determine how many points the revised team is expected to earn during the course of a full season. The projected point totals for the original team may be subtracted from the projected point totals of the revised team to determine a difference or delta. This delta is the value of this specific transaction to the specific team. This calculation can easily be performed for all combinations of players available to be added or dropped to locate the highest value roster changes available. Similar calculations could be used to determine which players to start and which to put on the bench.

Trade values could be calculated in much the same way. Because a quantitative resulting trade value can be computed for each team involved in the trade, it can help identify good or bad trades for an owner. It can also identify trades that are either even, or at least benefit all teams involved.

This methodology is also applicable to drafting of players prior to the start of a fantasy league. When drafting, each player has some value to each team, and that value changes with each pick, in accordance with the players already picked for that team, and the remaining pool of players from which a player may be selected. In some embodiments, the algorithm may assume that a particular owner’s team is populated with a lineup of average players with average statistics in each stat category for each position. The so-called average player for a particular position may be determined by averaging the statistics of all the players at that position that are expected to be drafted. The algorithm will then evaluate, for a possible draftee, how that draftee is predicted to impact the end of year points for the team. This may be accomplished by using the possible draftee’s predicted stats along with the predicted stats for the average players on the team, which can be assumed to perform in an average manner throughout the season. As players are actually drafted, that player’s stat predictions are used along with the stats of the remaining average players (the average player’s stats are changed to reflect the average stats of all those players remaining that are expected to be drafted), and new average stats (and points) for the team are recomputed.

**BRIEF DESCRIPTION OF THE DRAWINGS**

**FIG. 1** is an exemplary chart illustrating the rotisserie style scoring system in the context of a fantasy baseball league.

**FIG. 2** is an illustration of the exemplary selection of players for a team in a conventional fantasy league.

**FIG. 3** is an illustration of an exemplary fantasy league player evaluation system according to embodiments of the invention.

**FIG. 4** illustrates exemplary historical league stat and points data for a particular stat category according to embodiments of the invention.

**FIG. 5a** illustrates exemplary current league stat and points data for a particular stat category at a particular point in the season according to embodiments of the invention.

**FIG. 5b** illustrates exemplary current league stats and points data for a particular stat category extrapolated cut to a full season according to embodiments of the invention.

**FIG. 6** illustrates an exemplary computing system capable of implementing embodiments of the invention.

**FIG. 7** illustrates a typical computing system that may be employed to implement processing functionality in embodiments of the invention.

**DETAILED DESCRIPTION OF THE INVENTION**

Embodiments of the invention are directed to the evaluation of players in the context of a fantasy sports league. A player value, personalized for a specific fantasy team, is generated for a particular player being contemplated for addition to the starting lineup of that team. This value informs a team owner how that player is expected to affect the fantasy points earned by the team at the end of a season, and allows the owner to make managerial decisions optimized for earning the most fantasy points and winning the fantasy league.

The present invention is applicable to so-called “draft and trade” game types and rotisserie scoring types. More generally, embodiments of the invention are applicable to any scoring system that is not a stat modifier scoring type.

**FIG. 3** is an illustration of an exemplary fantasy league player evaluation system according to embodiments of the invention. Note that in embodiments of the invention as illustrated in FIG. 3, the conventional generic player rankings block has been replaced with a historical league stats and points block, a current league stats and points block, a player stat predictions block and a blending function block which blends the historical and current stats to predict how the addition of a player from an “other players” block in combination with the players currently in the team’s starting lineup (block) may impact the team’s overall points, based on that player’s stat predictions and the rest of the starting lineup’s stat predictions. The fantasy team owner may implicitly use this blending function to determine how the current lineup is to be adjusted, and optimize his revised lineup accordingly (block). These blocks will be discussed in further detail below.

The “other players” block is comprised of the pool of available players from which a team owner can select to insert a player into the starting lineup. The pool of available players can be players on the team but not in the starting
lineup ("on the bench"), players on other teams (that the owner can trade for), or free agents (players not on any team). Block 310 represents the players on the owner's team that have been selected for the starting lineup. Because only players in a team's starting lineup can accumulate stats, a fantasy team owner must consider how each of these players is contributing to the stat totals of the team.

When considering whether to add a player from the "other players" block 308 to the players in the current starting lineup (block 310), the projected stat totals for the prospective player and the players that would remain in the starting lineup must be considered for the remainder of the season. The player stat predictions block 304 provides the projected stats for these players. A player's projected stats may be calculated using any number of methodologies, and provided along with some measure of the predictions' expected accuracy (e.g., an accuracy range between an upper and lower bound). This may be accomplished using standard deviations or other well-known methodologies. It should be understood that even the method of generating stats (automated equations, hand editing, etc.) may affect the accuracy.

Player stat projections may be based on the length of time for which the projection is needed. For example, stat projections may be needed for an entire season, in which case their expected accuracy may be low as compared to stat projections needed for the remainder of an ongoing season, in which case the projections are likely to become more accurate as actual stats are accrued and the end of the season draws near.

Some stat projection algorithms may receive as input a database of player statistics from the past, and consider factors such as each player's trends over the course of a season, who that player is playing, whether the game is a day or night game, whether the pitcher is right or left handed, and/or whether the game will be played on real or artificial grass, etc., and generate predicted stats for that player for one or more games or the remainder of the season. Another method for generating stat projections is to run simulation engines, taking into account the factors mentioned above, simulate one or more games many times and compute average stats for the games. One example of a source for baseball player stat predictions is The Bill James Handbook 2007, ACTA Sports 2007, ISBN: 0879463112.

In the revised lineup block 312, one or more players in the current starting lineup of a team (block 310) may be replaced with a new player selected from the pool of free players in the "other players" block 308. Each potential unique combination of players that form the revised lineup will produce a different projected stat total for the team at the end of the season. The team stat totals for each stat category are calculated by adding the current team stat totals to the projected stats for the players in the revised lineup.

The key question to be evaluated when choosing a player to add to the starting lineup of a team is which combination of players (resulting in different team stat totals) will maximize the number of fantasy points earned by the team at the end of the season. To answer this question, team points must be optimized according to an optimization function of the form:

\[
\text{team points} = \text{func}(\text{stat1 total, stat2 total, stat3 total, ...})
\]

where \text{stat1 total, stat2 total, stat3 total, etc.} are the projected team stat totals for various stat categories as established by fantasy league rules. The result is the total team points projected to be earned by the team at the end of the season. The function may weight every stat category equally, in which case the team points may be computed as:

\[
\text{team points} = \text{A(points for stat1 total)} + \text{B(points for stat2 total)} + \text{C(points for stat3 total)}
\]

In another embodiment, more weight may be assigned to one stat category as compared to another, in which case the team points may be computed as:

\[
\text{team points} = A \times \text{points for stat1 total} + B \times \text{points for stat2 total} + C \times \text{points for stat3 total}
\]

where A, B, C etc. are weighting factors. These functions are defined based on fantasy league rules.

In order to compute these projected team point totals, embodiments of the invention may utilize historical trends and/or current league standings. As mentioned above, a database of historical league stats and points for teams is provided at block 300. This database may contain stats and points for teams in the same league during one or more previous seasons, or may contain stats and points for teams in different leagues operated by the same fantasy league management provider (e.g., Yahoo!) during one or more previous seasons, and may even contain stats and points for teams in other leagues operated by different fantasy league management providers during one or more previous seasons.

In some embodiments, data may be collected from past leagues with scoring rules similar to those in the target league. Data for the number of points and the totals for each stat category for each team from all of the reference leagues may be compiled. For a specific stat category, this data should resemble an "S" curve when plotted as team points earned on the horizontal axis, and team stat totals on the vertical axis. Polynomial regression may be used to create an approximation function for the data.

FIG. 4 illustrates exemplary historical league stat and points data for a particular stat category (e.g., HR) according to embodiments of the invention. In FIG. 4, teams with 150 HRs at the end of a season received 12 points on average, while teams with 100 HRs received 6 points on average. The curves appear smooth because they may represent the data from a number of different past league stats.

In addition to historical league stats, a database of the current league's stats and points for each team may be provided (see block 302 in FIG. 3) during the current season. FIG. 5a illustrates exemplary current league stat and points data for a particular stat category (e.g., HR) according to embodiments of the invention. FIG. 5a indicates that a team with 75 HRs at the current point in the season (e.g., halfway through the season) received 12 points, while a team with 50 HRs at the current point in the season received 6 points. However, before the current league stats and points data can be utilized or blended with the historical league stats and points data, the current stats and points data must be extrapolated to a full season. FIG. 5b illustrates the exemplary current league stats and points data extrapolated to a full season. Note that unlike the historical data, which may be smoother due to the inclusion and averaging of more historical data points, the current league stat and point data for each particular stat category will be a step function. It should also be understood that early in the season, the curves may be drawn like S curves. As the season progresses, the curves should appear more like S curves.
After the optimization function is created, the team points equation is established, and projected player stats are made available, the historical and extrapolated current league stats may be used to determine exactly how many points any given player is expected to contribute to a specific team. The historical and extrapolated current league stats may be blended to optimize their accuracy.

Before the current season starts, there will be no data for the current league stat and point charts. In other words, there will be no data and no step function of current league stats and points for any stat category. In that case, in some embodiments of the invention the historical league stat and points data will be used. Thus, for example, if an owner has chosen his starters for his lineup but is now considering revising the lineup, only FIG. 4 would be used to see how the addition of one player and/or the subtraction of another player is likely to affect the teams points for the HR category at the end of the season. If an owner’s current lineup projects to having 100 HRs at the end of the season, and the owner is able to drop a player A who is projected to hit 5 HRs in favor of a player B expected to hit 55 HRs, the historical chart can be used to show the owner that the proposed transaction should result in 6 additional points for the HR stat over the course of the full year. However, other stats may show that the dropping of player A in favor of player B may actually result in the loss of more than 6 points in the other stat categories over the course of the final year, and therefore would not make sense to make the transaction.

Of course, the historical league stats represent data from previous seasons or different leagues, and do not represent how the current league’s teams and players are performing, and how each team’s performance is actually translating into points. As noted above, as the season progresses, the extrapolated current league stat and point charts become more like an S curve and may become a more accurate predictor of how stats will translate to points at the end of the season. Therefore, as the season progresses, in some embodiments of the invention more and more weight may be given to the extrapolated current league stats and points charts, and less and less weight should be given to the historical league stats and points charts. Eventually, by the last game of the season, most or all of the weight should be given the extrapolated current league stats and points charts, and little or no weight should be given to the historical league stats and points charts. During the middle part of the season, the historical and extrapolated current league stats and points charts may be used about equally. The ratio of usage of historical and extrapolated current league stats may be referred to herein as the blending function. In one embodiment, the ratio or percentage of usage of historical league data varies linearly from 100% to 0% as the season progresses from beginning to end, while the ratio of usage of extrapolated current league data varies linearly from 0% to 100% as the season progresses from beginning to end.

In addition to this linear weighting, weighting can be modified based on the accuracy of the stat predictions, if the accuracy can be determined. In other words, if the stat predictions were 100% accurate, then at any point in the season, the current league stats and points may be used exclusively. The formula might be: team points=(100-prediction accuracy)\( f_{\text{historical stats}}(\text{stats}) + \text{prediction accuracy} \times f_{\text{current stats}}(\text{stats}) \), where prediction accuracy ranges from 0 to 100 and is a function of at least the percent of the season completed. If the season is 70% complete, \( f_{\text{historical stats}}(\text{stats}) \) is a function representing historical team points as a function of stats for all stat categories, and \( f_{\text{current stats}}(\text{stats}) \) is a function representing current team points as a function of stats for all stat categories.

In other embodiments of the invention, however, only the historical league stats and points data may be used to predict the total team points at the end of the season.

With player stat predictions, historical league stats and points, and extrapolated current league stats and points data available, embodiments of the invention may be utilized to evaluate whether a fantasy team owner should add one or more players to the starting lineup of the team and drop one or more other players. To do this, player stat predictions are first used to predict end-of-season stats for the current starting lineup of a team. Historical and/or extrapolated current league stat/point functions (generally referred to herein as stats-to-points conversion data) may then be used to determine how many points the team is expected to earn during the course of a full season based on the predicted end-of-season stats. The historical and extrapolated current league stats may be utilized in accordance with a blending function, which gives more weight to historical league stats early in the current season, and more weight to extrapolated current league stats later in the season. Next, projected stats for the player or players under consideration for being dropped may be subtracted from the current team predicted end-of-season stats, and the projected stats of the player or players under consideration for being added to the current team may be added to the current team predicted end-of-season stats. These revised team stat totals may once again be applied to the historical and extrapolated current league stat/point functions to determine how many points the revised team is expected to earn during the course of a full season. The projected point totals for the original team may be subtracted from the projected point totals of the revised team to determine a difference or delta. This delta is the value of this specific transaction to this specific team. This calculation can easily be performed for all combinations of players available to be added or dropped to locate the highest value roster changes available. Similar calculations could be used to determine which players to start and which to put on the bench.

Trade values could be calculated in much the same way. Because a quantitative resulting trade value can be computed for each team involved in the trade, it can help identify good or bad trades for an owner. It can also identify trades that are either even, or at least benefit all teams involved.

Embodiments of the invention are also applicable to drafting of players prior to the start of a fantasy league. When drafting, each player has some value to each team, and that value changes with each pick, in accordance with the players already picked for that team, and the remaining pool of players from which a player may be selected. In some embodiments, the algorithm may assume that a particular owner’s team is populated with a lineup of average players with average statistics in each stat category for each position. The so-called average player for a particular position is determined by averaging the statistics of all the players at that position that are eligible to be drafted. The algorithm will then evaluate, for a possible draftee, how that draftee is predicted to impact the end of year points for the team. This may be accomplished by using the possible draftee’s predicted stats along with the predicted stats for the average players on the team, which can be assumed to perform in an average manner throughout the season. As players are actually drafted, that player’s stat predictions are used along with the stats of the
remaining average players (the average player’s stats are
to reflect the average stats of all those players
remaining that are eligible to be drafted), and new average
stats (and points) for the team are recomputed.

[0042] FIG. 6 illustrates an exemplary computing system
capable of implementing embodiments of the invention. Fantas
ty league management providers may employ fantasy
league servers 616 with software applications that allow
league owners to create fantasy leagues with configurable
league rules and manage those leagues, and allow fantasy
league owners to join leagues, draft players and make roster
changes. When a league and teams are created, the necessary
storage may be created for the league and teams (e.g. storage
602, 604 or 614, described hereinafter). Processes may be
automatically performed on a periodic basis to obtain the
statistics for the day, apply them to each team, and recalculate
points.

[0043] In FIG. 6, a transaction evaluation engine or logic
600 may be implemented within the fantasy league server 616
and utilized to predict the total points a team (including one or
more prospective players) may earn at the end of a season, and
in some embodiments perform the blending function
described above. The transaction evaluation logic 600 may
comprise one or more processors configured for implemen-
ting blending function code, or any computing device, includ-
ing dedicated hardware or state machines. Current fantasy
league data; historical fantasy league data, and player stat
predictions, each generally representing a database, may be
stored in storage or other memory 602, 604 and 614, respec-
tively. Storage 602, 604 and 614 may be separate or com-
bined, and may be co-located with the transaction evaluation
logic or located in other servers. A web server 606, which
may be physically separate from or co-located with the fantasy
league server 616, is communicatively coupled to the trans-
action evaluation logic 600 and fantasy league data storage
602 and employed to interact with a user or client 608
and enable the user to input proposed current lineup changes
and receive points predictions as computed by the transaction
evaluation logic. An owner or user 608, through the user’s
web browser 610 being executed on a computer 612, can
interact with the web server 604 to draft players, evaluate
proposed lineup changes, and make changes to the fantasy
team. The transaction evaluation logic 600 may receive cur-
rent fantasy league data, historical fantasy league data and the
player stat predictions from storage 602, 604 and 614 to
determine how many team points a player being considered
for addition to a team should be expected to contribute at the
end of the season, accounting for the corresponding dropping
of a player. This data may be co-located with the processors
or located in other servers.

[0044] While embodiments of the invention have been
described in terms of particular embodiments and illustrative
figures, those of ordinary skill in the art will recognize that
the embodiments of the invention are not limited to the embody-
ments or figures described. Although embodiments of the
invention are described, in some instances, in the context of
fantasy baseball, those skilled in the art will recognize that
such terms are also used in a generic sense herein, and that
embodiments of the invention are not limited to such systems.

[0045] Those skilled in the art will recognize that the opera-
tions of the various embodiments may be implemented using
hardware, software, firmware, or combinations thereof, as
appropriate, or in some embodiments may be implemented all
or in part by a user performing the operations manually. For
example, some processes can be carried out using processors
or other digital circuitry under the control of software, firm-
ware, or hard-wired logic. (The term “logic” herein refers to
fixed hardware, programmable logic and/or an appropriate
combination thereof, as would be recognized by one skilled
in the art to carry out the recited functions.) Software and firm-
ware can be stored on computer-readable media. Some other
processes can be implemented using analog circuitry, as is
well known to one of ordinary skill in the art. Additionally,
memory or other storage, as well as communication compo-
nents, may be employed in embodiments of the invention.

[0046] FIG. 7 illustrates a typical computing system 700
that may be employed to implement processing functionality
in embodiments of the invention. Computing systems of this
type may be used in the fantasy league server and web server,
for example. Those skilled in the relevant art will also recog-
nize how to implement embodiments of the invention using
other computer systems or architectures. Computing system
700 may represent, for example, a desktop, laptop or note-
book computer, hand-held computing device (personal digi-
tal assistant (PDA), cell phone, palmtop, etc.), mainframe,
server, client, or any other type of special or general purpose
computing device as may be desirable or appropriate for a
given application or environment. Computing system 700 can
include one or more processors, such as a processor 704.
Processor 704 can be implemented using a general or special
purpose processing engine such as, for example, a micropro-
cessor, microcontroller or other control logic. In this example,
processor 704 is connected to a bus 702 or other communica-
tions medium.

[0047] Computing system 700 can also include a main
memory 708, such as random access memory (RAM) or other
dynamic memory, for storing information and instructions
to be executed by processor 704. Main memory 708 also may
be used for storing temporary variables or other intermediate
information during execution of instructions to be executed
by processor 704. Computing system 700 may likewise
include a read only memory (ROM) or other static storage
device coupled to bus 702 for storing static information
and instructions for processor 704.

[0048] The computing system 700 may also include informa-
tion storage system 710, which may include, for example,
a media drive 712 and a removable storage interface 720. The
media drive 712 may include a drive or other mechanism to
support fixed or removable storage media, such as a hard
disk drive, a floppy disk drive, a magnetic tape drive, an optical
disk drive, a compact disk (CD) or digital versatile disk
(DVD) drive (R or RW), or other removable or fixed media
drive. Storage media 718, may include, for example, a hard
disk, floppy disk, magnetic tape, optical disk, CD or DVD,
or other fixed or removable medium that is read by and written
to by media drive 714. As these examples illustrate, the storage
media 718 may include a computer-readable storage medium
having stored therein particular computer software or data.

[0049] In alternative embodiments, information storage
system 710 may include other similar components for allow-
ing computer programs or other instructions or data to be
loaded into computing system 700. Such components may
include, for example, a removable storage unit 722 and an
interface 720, such as a program cartridge and cartridge inter-
face, a removable memory (for example, a flash memory
or other removable memory module) and memory slot, and
other removable storage units 722 and interfaces 720 that
allow software and data to be transferred from the removable storage unit 718 to computing system 700. 0050 Computing system 700 can also include a communications interface 724. Communications interface 724 can be used to allow software and data to be transferred between computing system 700 and external devices. Examples of communications interface 724 can include a modem, a network interface (such as an Ethernet or other network interface card (NIC)), a communications port (such as for example, a USB port, a PCMCIA slot and card, etc. Software and data transferred via communications interface 724 are in the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communications interface 724. These signals are provided to communications interface 724 via a channel 728. This channel 728 may carry signals and may be implemented using a wireless medium, wire or cable, fiber optics, or other communications medium. Some examples of a channel include a phone line, a cellular phone link, an RF link, a network interface, a local or wide area network, and other communications channels.

0051 In this document, the terms “computer program product,” “computer-readable medium” and the like may be used generally to refer to media such as, for example, memory 708, storage device 718, or storage unit 722. These and other forms of computer-readable media may store one or more instructions for use by processor 704, to cause the processor to perform specified operations. Such instructions, generally referred to as “computer program code” (which may be grouped in the form of computer programs or other groupings), when executed, enable the computing system 700 to perform functions of embodiments of the invention. Note that the code may directly cause the processor to perform specified operations, be compiled to do so, and/or be combined with other software, hardware, and/or firmware elements (e.g., libraries for performing standard functions) to do so.

0052 In an embodiment where the elements are implemented using software, the software may be stored in a computer-readable medium and loaded into computing system 700 using, for example, removable storage drive 714, drive 712 or communications interface 724. The control logic (in this example, software instructions or computer program code), when executed by the processor 704, causes the processor 704 to perform the functions of embodiments of the invention described herein.

0053 It will be appreciated that, for clarity purposes, the above description has described embodiments of the invention with reference to different functional units and processors. However, it will be apparent that any suitable distribution of functionality between different functional units, processors or domains may be used without detracting from embodiments of the invention. For example, functionality illustrated to be performed by separate processors or controllers may be performed by the same processor or controller. Hence, references to specific functional units are only to be seen as references to suitable means for providing the described functionality, rather than indicative of a strict logical or physical structure or organization.

0054 Although embodiments of the invention have been described in connection with some particular embodiments, it is not intended to be limited to the specific form set forth herein. Rather, the scope of embodiments of the invention is limited only by the claims. Additionally, although a feature may appear to be described in connection with particular embodiments, one skilled in the art would recognize that various features of the described embodiments may be combined in accordance with embodiments of the invention.

0055 Furthermore, although individually listed, a plurality of means, elements or method steps may be implemented by, for example, a single unit or processor. Additionally, although individual features may be included in different claims, these may possibly be advantageously combined, and the inclusion in different claims does not imply that a combination of features is not feasible and/or advantageous. Also, the inclusion of a feature in one category of claims does not imply a limitation to this category, but rather the feature may be equally applicable to other claim categories, as appropriate.

1. An apparatus for determining a value of proposed changes to a starting lineup of a current fantasy team in a fantasy sports league, comprising:

   a. Transaction evaluation logic operable for receiving current fantasy team data and a proposed starting lineup change,

   b. Computing predicted end-of-season stats for both the current fantasy team and a proposed changed team based upon the proposed starting lineup change using player stat predictions,

   c. Computing predicted end-of-season point totals for both the current fantasy team and the proposed changed team using stats-to-points conversion data, and

   d. Computing a difference between the predicted end-of-season point totals for both the current fantasy team and the proposed changed team, the difference representing the value of the proposed starting lineup change.

2. The apparatus of claim 1, the stats-to-points conversion data comprising historical fantasy league stats and points data providing a conversion from stats to points based on data from previous fantasy leagues.

3. The apparatus of claim 2, the stats-to-points conversion data comprising extrapolated current fantasy league stats and points data providing a conversion from stats to points based on data from the current fantasy league, and wherein the computing of the predicted end-of-season point totals for both the current team and the proposed changed team is performed using a blending function that utilizes a weighting of the historical fantasy league stats and points data and the extrapolated current fantasy league stats and points data.

4. The apparatus of claim 3, the blending function comprising linearly weighting the historical fantasy league stats and points data from 100% to 0% and linearly weighting the extrapolated current fantasy league stats and points data from 0% to 100% as a season progresses from start to finish.

5. The apparatus of claim 2, the historical fantasy league stats and points data comprising stats and points for teams in the same league during one or more previous seasons.

6. The apparatus of claim 2, the historical fantasy league stats and points data comprising stats and points for teams in different leagues operated by the same fantasy league management provider during one or more previous seasons.

7. The apparatus of claim 2, the historical fantasy league stats and points data comprising stats and points for teams in other leagues operated by different fantasy league management providers during one or more previous seasons.

8. The apparatus of claim 1, the proposed changes to the starting lineup of the current fantasy team comprising a proposed drafting of a player prior to a start of a season, the
starting lineup initially including a team of fictitious players having stats representing average stats for all players available to be drafted.

9. The apparatus of claim 1, the stats-to-points conversion data including current fantasy league stats and points data, the apparatus comprising a current fantasy league database communicatively coupleable to the transaction evaluation logic and operable for providing current fantasy team data including current team rosters and the current fantasy league stats and points data for all teams in the league.

10. The apparatus of claim 1, the stats-to-points conversion data including historical fantasy league stats and points data, the apparatus comprising a historical fantasy league database communicatively coupleable to the transaction evaluation logic and operable for providing the historical fantasy team stats and points data.

11. The apparatus of claim 1, comprising a player stat predictions database communicatively coupleable to the transaction evaluation logic and operable for providing the player stat predictions for a remainder of a current season.

12. The apparatus of claim 1, comprising a web server communicatively coupleable to the transaction evaluation logic and operable for receiving proposed current lineup changes from a fantasy team owner and sending transaction values back to the fantasy team owner as computed by the transaction evaluation logic.

13. A server system comprising the apparatus and web server of claim 12, the server system coupleable to a storage area network.

14. A method for determining a value of proposed changes to a starting lineup of a current fantasy team in a fantasy sports league, comprising:

using player stat predictions to determine total stats expected to be compiled at an end of a season for the starting lineup, and using stats-to-points conversion data to compute expected points to be accrued at the end of the season for the starting lineup based on the total stats expected to be compiled;

using player stat predictions to determine first total stats compiled at the end of the season for a first player being considered for addition to the current starting lineup;

using player stat predictions to determine second total stats compiled at an end of a season for a second player being considered for removal from the current starting lineup;

using the stats-to-points conversion data to compute the expected points to be accrued at the end of the season for the proposed changed lineup based on the total stats expected to be compiled plus the first total stats and less the second total stats; and

computing a difference between the expected points to be accrued at the end of the season for the proposed changed lineup and the expected points to be accrued at the end of the season for the starting lineup, the difference representing the value of the proposed changes.

15. The method of claim 14, the stats-to-points conversion data comprising historical fantasy league stats and points data providing a conversion from stats to points based on data from previous fantasy leagues.

16. The method of claim 15, the stats-to-points conversion data comprising current fantasy league stats and points data providing a conversion from stats to points based on data from the current fantasy league, and wherein the computing of the expected points to be accrued at the end of the season for both the starting lineup and the proposed changed lineup is performed using a blending function that utilizes a weighting of the historical fantasy league stats and points data and the current fantasy league stats and points data.

17. The method of claim 16, the blending function comprising linearly weighting the historical fantasy league stats and points data from 100% to 0% and linearly weighting the current fantasy league stats and points data from 0% to 100% as a season progresses from start to finish.

18-20. (canceled)

21. The method of claim 14, the proposed changes to the starting lineup of the current fantasy team comprising a proposed drafting of a player prior to a start of a season, the starting lineup initially including a team of fictitious players having stats representing average stats for all players available to be drafted.

22. The method of claim 14, the stats-to-points conversion data including current fantasy league stats and points data, the method comprising accessing a current fantasy league database to obtain current fantasy team data including current team rosters and the current fantasy league stats and points data for all teams in the league.

23. The method of claim 14, the stats-to-points conversion data including historical fantasy league stats and points data, the method comprising accessing a historical fantasy league database to obtain the historical fantasy team stats and points data.

24. The method of claim 14, comprising accessing a player stat predictions database to obtain the player stat predictions for a remainder of a current season.

25. A computer-readable medium comprising program code for determining a value of proposed changes to a starting lineup of a current fantasy team in a fantasy sports league, the program code for causing performance of a method comprising:

using player stat predictions to determine total stats expected to be compiled at an end of a season for the starting lineup, and using stats-to-points conversion data to compute expected points to be accrued at the end of the season for the starting lineup based on the total stats expected to be compiled;

using player stat predictions to determine first total stats compiled at the end of the season for a first player being considered for addition to the current starting lineup;

using player stat predictions to determine second total stats compiled at an end of a season for a second player being considered for removal from the current starting lineup;

using the stats-to-points conversion data to compute the expected points to be accrued at the end of the season for the proposed changed lineup based on the total stats expected to be compiled plus the first total stats and less the second total stats; and

computing a difference between the expected points to be accrued at the end of the season for the proposed changed lineup and the expected points to be accrued at the end of the season for the starting lineup, the difference representing the value of the proposed changes.

26. The computer-readable medium of claim 25, the stats-to-points conversion data comprising historical fantasy league stats and points data providing a conversion from stats to points based on data from previous fantasy leagues.

27. The computer-readable medium of claim 26, the stats-to-points conversion data comprising current fantasy league stats and points data providing a conversion from stats to points based on data from the current fantasy league, and
wherein the computing of the expected points to be accrued at the end of the season for both the starting lineup and the proposed changed lineup is performed using a blending function that utilizes a weighting of the historical fantasy league stats and points data and the current fantasy league stats and points data.

28. The computer-readable medium of claim 27, the blending function comprising linearly weighting the historical fantasy league stats and points data from 100% to 0% and linearly weighting the current fantasy league stats and points data from 0% to 100% as a season progresses from start to finish.

29-35. (canceled)

* * * * *