

A Solution for the DMcommunity July-2022 Challenge

“Evaluate Team Performance”

<https://dmcommunity.org/challenge/challenge-july-2022>

Using the Executable English Platform

www.executable-english.com

Adrian Walker

1. THE EXECUTABLE ENGLISH LANGUAGE

The problem is represented in the language Executable English (EE). Here is a simplified description of part of EE.

An EE statement is an EERule, or an EETable.

An EERule consists of one or more EELines, followed by an underline (---), followed by an EELine.

An EETable is an EELine followed by a double underline (==) followed by a relational table (which may be empty)

An EELine is a sentence in English (or other language) containing one or more place holders (variables) marked by tags such as: **this- that- a- an-**.

With a few exceptions, an EELine is open vocabulary and open syntax.

In a valid EE application, every line is either a first line of an ETable, or a last line of an ERule, or a predefined line. (recursion is supported)

The EE platform checks incoming text typed into a web page against the above description, and produces messages to guide the author.

2. WRITING THE PROBLEM IN EXECUTABLE ENGLISH

Hint -- when working with the platform, keep your favorite editor open in a second window, and copy-paste in both directions between it and the browser page.

Here are two ETables copied from the problem description and supplied with invented header lines:

```
player who plays game with efficiency this-eff gets this-number points
```

```
=====
```

best	5
better	3
good	2
bad	-2
worst	-5

```
for this-team this-player on this-date had efficiency this-e
```

```
=====
```

Mustungs	Brown	4/1/2022	good
Mustungs	Brown	4/2/2022	better
Eagles	Black	4/1/2022	good
Eagles	Black	4/2/2022	better

Here is an EERule to implement the fact that the problem statement requires results about points, not about efficiency:

```
for some-team some-player on some-date had efficiency some-e
player who plays game with efficiency that-e gets some-number points
-----
for that-team that-player on that-date had that-number points
```

Here is an EERule making use of a syntactical item, (sum) that is built in to the platform

```
some-team is a team
sum a-number : for that-team some-player on some-date had some-number points = a-total
-----
that-team had that-total points
```

Here is an EERule making use of another syntactical item (not:) that is built in

```
some-team had some-total points
not : there is a team that had more than that-total points
-----
that-team had the most points
```

If we type or copy-paste the above EERules and EETables into a web page, the platform suggests that we still need EERules or tables for the following two lines:

```
some-team is a team and
there is a team that had more than some-total points
```

Typing in the following two rules completes the solution.

```
for some-team some-player on some-date had efficiency some-e
```

```
-----
```

```
that-team is a team
```

```
some-team had some-total points
```

```
that-total is greater than some-number
```

```
-----
```

```
there is a team that had more than that-number points
```

3. RESULTS FROM RUNNNING THE PROBLEM

Note that we only copy-pasted part of the problem data above, so the result is that both teams scored 5. Adding the full problem data in the two tables, we get the result

```
this-team had the most points
```

```
=====
```

```
Mustung
```

(continued)

4. GETTING AN EXPLANATION OF THE RESULT

The system automatically generates a hypertexted [explanation](#) showing the steps it made in finding the result:

Mustungs had 11 points

not: there is a team that had more than 11 points

Mustungs had the most points

Mustungs is a team

sum eg-number : for Mustungs eg-player on eg-date had eg-number points = 11

Mustungs had 11 points

for Mustungs Brown on 4/1/2022 had efficiency good

Mustungs is a team

for Mustungs Brown on 4/1/2022 had 2 points

for Mustungs Brown on 4/2/2022 had 3 points

...

...

```
for Mustungs Brown on 4/1/2022 had 2 points
for Mustungs Brown on 4/2/2022 had 3 points
for Mustungs Brown on 4/3/2022 had 5 points
for Mustungs Robinson on 4/1/2022 had -5 points
...
-----
sum eg-number : for Mustungs eg-player on eg-date had eg-number points = 11

...

...
for Mustungs Smith on 4/3/2022 had efficiency bad
player who plays game with efficiency bad gets -2 points
-----
for Mustungs Smith on 4/3/2022 had -2 points
and so on.
```

(continued

5. HOW THE RESULTS WERE PRODUCED

The EE platform uses a rule engine designed to assign a highly declarative meaning to a set of rules and tables. In particular, authors need not be concerned with rule firing sequences -- shuffling the rules and tables does not change the results.

The engine combines forward and back-chaining of rules, automatically switching between the two methods. There are extensive publications about the EE platform, as well as the live online platform itself, at www.executable-english.com. You can view, edit and run the Mustangs/Eagles problem by pointing a browser to the site and choosing DMcommunity1 . Please see detailed steps at <https://www.executable-english.com/quickstart.html>

Hint -- when working with the platform, keep your favorite editor open in a second window, and copy-paste in both directions between it and the browser.

Version Aug 16 2022