

# **Homosapien Modeling Language (HML)**

## **Language Proposal**

COMS W4115: Programming Languages and Translators

Professor Stephen A. Edwards

Computer Science Department

Summer 2008 Columbia University

Dated: 2008-06-03

Derek Ng

[dn2150@columbia.edu](mailto:dn2150@columbia.edu)

Columbia Video Network

## Objective

The Homosapien Modeling Language (HML) is a language meant to allow programmers the ability to model the nature of human beings. With HML, a programmer could model the effects of external forces on the human body.

## Background

Humans in general are complex creatures. Scientists have spent years researching different aspects of the human body. They have tried to generate the timeline of changes that occur during birth, life, and death. To aide in such studies, I will attempt to create an Object Orientated language that will assist in this research by creating a flexible language to adapt over time.

## Features

All humans go through 3 main stages of life. They are born, live for some duration of time, and ultimately die. Each human has vital (height, weight, age, happiness) and non-vital (name, sex, race) statistics. While the human is living, the vital statistics fluctuate in some range. If some external force causes these statistics to fluctuate to some extreme, the human will simply die. The external force feature of the programming language will be generic as to allow programmers to introduce their own forces for their models.

## Language Overview

Programmers will be able to define elements: humans, statistics, and forces. These elements will be referred to by alphabetic strings. These strings are case in-sensitive.

Each statistic will be related to a human and/or force. The relation will be denoted by appending a period and force name to the human or force that is describes.

Programmers can begin comments with `/*` and end them with `*/`.

## Code Samples

Note: I'm learning Java as speak for this class. I am writing the code samples as a best guess how things would appear in HML, but may appear different after implementing in Java.

```
/* Human is born */
```

```
Born Human Derek = new Human;
```

```
/* Set Human attributes, artificially aged */
```

```
Derek.height = 400;
```

```
Derek.age = 10;
```

```
Derek.weight = 410;
```

```
Derek.happiness = 80;
```

```

/* Create external force */
Create Force Burger = new Force;

/* Set Force attributes */
Burger.weight = 10;
Divorce.happiness = -100;

/* Apply Force + Display effect */
Derek+Burger;
Display(Derek);
Derek+Divorce;
Display(Derek);

```

Eventually, the divorce would set Derek's happiness below 0 which would trigger his death.

## Reserved Keywords

If  
 Else  
 While  
 Do  
 Human  
 Create  
 Display  
 TRUE  
 FALSE

## Operators

+ (applies a force)  
 = (assigns a statistic a value)  
 if (<statement>) { <statement> } else { <statement> }  
 while (<statement>) do { <statement> }

## Testing

A human will be created and exposed to several effects. I hope to create visuals of the effects.

## Further Expansion Concepts

In the longer term, I hope to create associations between humans so that communities of interested can be singled out for testing. For example, I would envision associating humans to a city habitat that had its own attributes (smog, congestion, small real estate) to see how it affects their statistics (health, happiness). On top of that, I would hope to develop forces that could affect a whole association (i.e. A

heat wave hits a city which causes multiple people to react differently. Maybe the older people have a heat stroke or the younger people get happy because they like the sun).

## References

**SLAG:** Structured Language Applied to Gaming