

## Description

### 1. using CLOS

Almost all of AGAST--story segments, goals, actions, etc.--is written using CLOS [Bobrow et al 87], an object oriented programming language built on Common Lisp [Steele et al 84].

Like other object oriented languages, CLOS has a hierarchy of inheritance, with leaf classes inheriting everything defined in classes along the path from the root to the leaf. Information storage is in slots. There are two types of slot inheritance in CLOS: class and instance. Class slots have their value inherited by all their subclasses; that is, the slot is defined in a superclass, and all subclasses access the same slot. Instance slots are passed down the hierarchy, with each instance getting its own copies of instance slots; thus each instance can have its own value for an inherited instance slot.

The class hierarchies deemed necessary for AGAST required the implementing of a third type of slot inheritance, the template slot. Template slots are inherited like instance slots, but at the leaf level they are turned into class slots so that instances all share the same slot value. For instance, the template slot NUM-OF-LEGS is defined in the class ANIMAL and is inherited by all subclasses. It is turned into a class slot in the leaf classes HUMAN and HORSE. All instances of HUMAN share the value of "2" for that slot, while all instances of HORSE share the value of "4."

Methods allow slots needing values to access other slots to determine or compute those values. For instance, each instance of a living thing has a slot containing instances of LEG. The number of LEGs is based on the value of the slot NUM-OF-LEGS of the class of the living thing. For EARTH-CREATUREs, the slot NUM-OF-LEGS is predefined; for ALIENS, the slot is a random number (with an upper limit, fortunately).

