Contract

Description

This ”contract” contains everything that you can assume about the inputs to the various modules of the model, as well as what is required of you on output. This is to attempt to ensure that the model is completely modular, and any piece in one of the categories can be substituted for any other piece in that category and the model would still run (hopefully this will lead to complex emergent interactions between the modules). Anything not specified are things that you cannot be ”sure” of, so you should not use them (unless you update the contract). However, creating and using variables within sections is fine, as long as other sections do not modify them. It is also helpful to look directly at the examples already in each folder to get an idea of requirements.

Ballot

Contains the code necessary to create the ballot and run the model (i.e. an experiment). The function vote with parameters (realtime use-model visible) is called by combine.lisp.

Memory

Contains any number of declarative memory entries (lisp s-expressions). Can make use of chunk types in combine.lisp.

Macronavigation

Contains the productions for navigating from race to race. One needs to make sure that the manual buffer is clear when entering the first production, as otherwise logging may not be finished.

Goal
Input:
State: Start-voting, Find-race
Left: Left of last race, -1 if start-voting
Right: Right of last race, -1 if start-voting

Output:
State: Storing-race-group, End
Left: Left of current race

Visual Location
Output: a race-header location in the visual location buffer

Visual
Input: attention somewhere in the last race (although it seems that it may be alright if this buffer is cleared as the model still works but best to be safe)

Imaginal
Input: the race, candidate, and party groups of the last race, in their respective fields (a full imaginal buffer from the race just voted on)
Output: an imaginal buffer with none in each of the group fields

Encoding

Contains the productions for encoding the race, party, and candidate groups for a specific race.

Goal
Input:
State: Storing-race-group
Left: Left of current race

Output:
State: Ready-to-make-choice
Left: Left of current race
Right: Right of current race

Visual Location
Input: a race-header location in the visual location buffer

Imaginal
Input: an imaginal buffer with none in each of the group fields
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in

Micronavigation

Contains the productions for finding the candidate to vote for within this race. Either sends the model to look for a new race if abstaining (find-next-race in the goal module), or attends a candidate and prepares to vote for it (moved-to-candidate in the goal module).

Goal
Input:
State: Ready-To-Make-Choice
Left: Left of race
Right: Right of race

Output:
State: Moved-to-candidate, Find-next-race
Left: Left of race
Right: Right of race

Visual
Output: attention on the candidate or party one wants to vote for, or in the case of abstaining attention somewhere in the current race

Imaginal
Input: an imaginal buffer with race, candidate, and party groups for the current race filled in
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in

Click

Contains the productions for clicking on the bubble corresponding to the candidate to vote on.

Goal
Input:
State: Moved-to-candidate
Left: Left of race
Right: Right of race

Output:
State: Find-next-race
Left: Left of race
Right: Right of race

Visual
Input: attention on the candidate or party one wants to vote for
Output: attention somewhere in the race (it could be in the same position as the beggining)

Imaginal
Input: an imaginal buffer with race, candidate, and party groups for the current race filled in
Output: an imaginal buffer with race, candidate, and party groups for the current race filled in