Heather Friedberg
10/13/11

CS 2310 – Multimedia Software Engineering

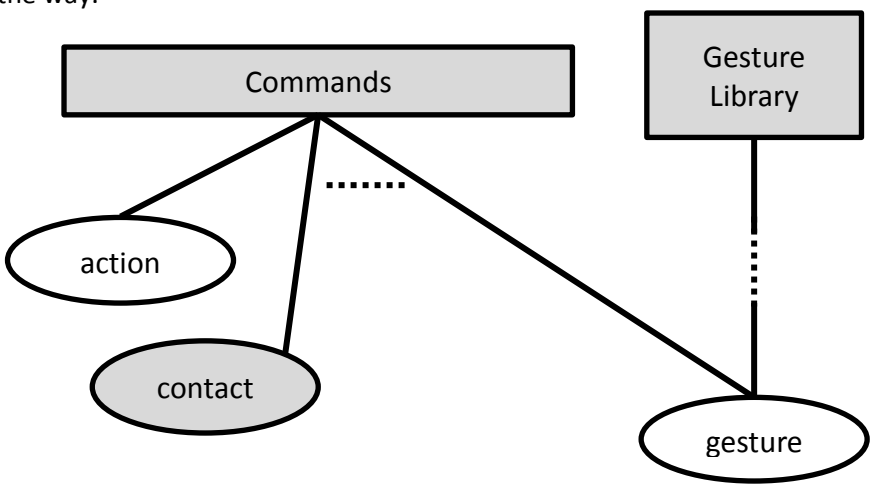
Exercise 4

In this exercise I present a multimedia functional database for a personal healthcare system. This system allows a user (senior citizen) to access a variety of multimedia documents using gestures. The Kinect interface is used to read in these gestures, these gestures are mapped into the space of gesture language, and translated into a query that will ultimately give the user the result he was looking for.

For the course of this exercise, we will assume that the system can recognize three basic gestures, each corresponding to an action. These three gestures are as follows:

Gesture	Short Meaning	Further Description
Touch hand to mouth	Food	The user would like someone to deliver them groceries
Hold hands in front (like on a wheel)	Drive	The user would like to go somewhere, and is requesting a car service
Wave hand in air	Help	The user is sick or has some other emergency, and needs help right away

Within the system, each of these gestures is part of a gesture library that the Kinect can recognize. In addition, there also needs to be a database of commands that the personal health system can act on and the corresponding actions, contact people, etc. for each of these. Commands is a relation that includes the attributes action (the actual action that needs to be taken, for example, call, visit, etc) and gesture, among other things like the contact person (who needs to be told to do the action). Gestures are most likely recorded as BLOBs. You can imagine that other multimedia documents, and therefore BLOBs, could also be attributes of command – for example, if there was a video monitor set up in the user's house, the system, on receiving a command, could play a video telling the user that the gesture had been received and what they should do. This could be something simple like "your groceries will arrive in ___ minutes" or in the case of a help command a video telling them to remain calm because help was on the way.



Let us assume that the Kinect interface uses a similarity metric GESTURECODE to determine the similarity of two gestures, most likely by measuring different coordinates of objects in its viewing scope and comparing them across the two gestures. There is also a similarity measure for the action; though this is much more straight-forward and the actions are probably from a small set and can be compared for equality directly, we will refer to this as ACTIONCODE.

$$GESTURE_{GESTURECODE(t1)} \rightarrow ACTION_{ACTIONCODE(t2)}$$

Because this relationship exists, there is a type-M dependency between action and gesture. You can imagine that our database of actions will include multiple examples of possible realizations of the gesture "food", for example, so that if multiple people do it a little differently there is more to compare it too.

The following are IC cards, used to describe the patterns in the system:

IC Card IC Name: Receive Gesture

Description: System Receives Gesture through the Kinect Interface

Interaction Pattern:

Quiet State | By Myself no Interaction | By Myself with Interaction | By Others no Interaction | By Others with Interaction | Mixed

My Task: Receive Gesture

Time Critical Condition: Need to be ready to receive at any time.

Name of Other IC: Translate Gesture

Message to Other IC: The gesture to translate

Other IC's Task: Translate the gesture to an action

Card 1 of 1 (If necessary please use several IC cards to describe an IC)

IC Card IC Name: Translate Gesture

Description: Translate Gesture to Action using Similarity Metric

Interaction Pattern:

Quiet State | By Myself no Interaction | By Myself with Interaction | By Others no Interaction | By Others with Interaction | Mixed

My Task: Translate the Gesture

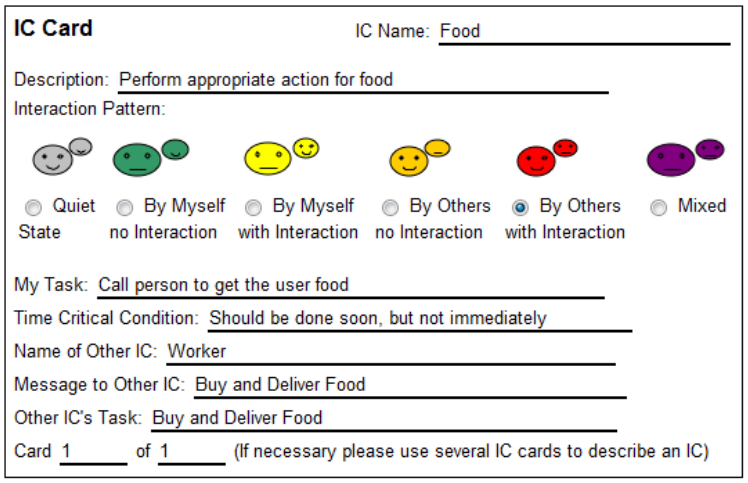
Time Critical Condition: Needs to be done fast - could be an emergency

Name of Other IC: Other parts of the system

Message to Other IC: Help translate

Other IC's Task: Help translate

Card 1 of 1 (If necessary please use several IC cards to describe an IC)



IC Card IC Name: Food

Description: Perform appropriate action for food

Interaction Pattern:

Quiet State | By Myself no Interaction | By Myself with Interaction | By Others no Interaction | By Others with Interaction | Mixed

My Task: Call person to get the user food

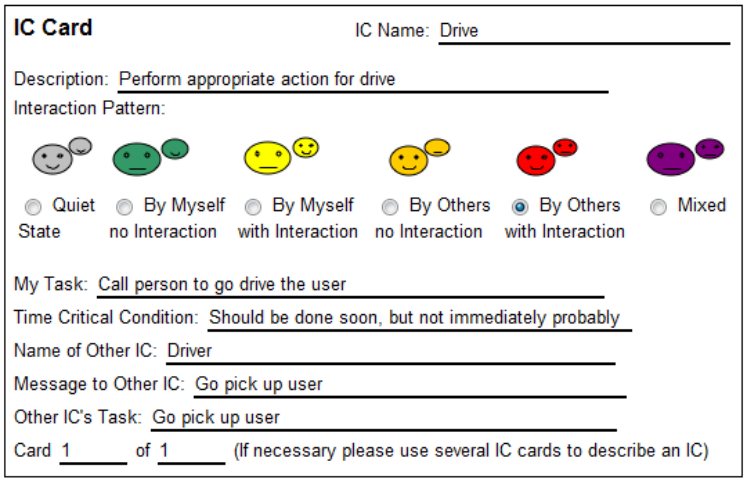
Time Critical Condition: Should be done soon, but not immediately

Name of Other IC: Worker

Message to Other IC: Buy and Deliver Food

Other IC's Task: Buy and Deliver Food

Card 1 of 1 (If necessary please use several IC cards to describe an IC)



IC Card IC Name: Drive

Description: Perform appropriate action for drive

Interaction Pattern:

Quiet State | By Myself no Interaction | By Myself with Interaction | By Others no Interaction | By Others with Interaction | Mixed

My Task: Call person to go drive the user

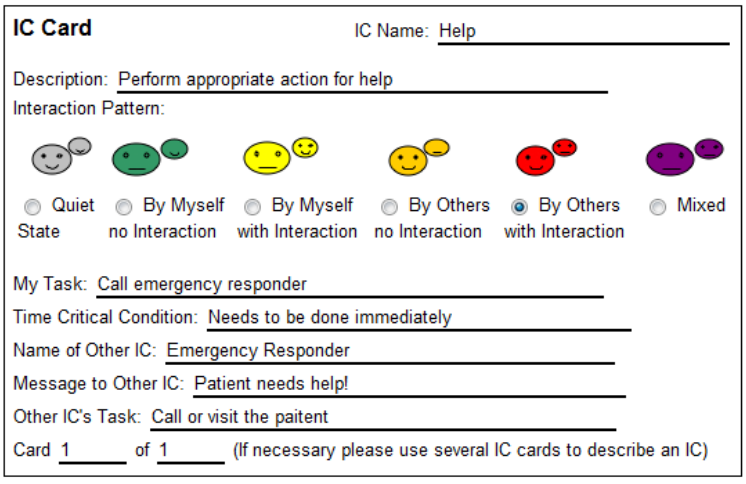
Time Critical Condition: Should be done soon, but not immediately probably

Name of Other IC: Driver

Message to Other IC: Go pick up user

Other IC's Task: Go pick up user

Card 1 of 1 (If necessary please use several IC cards to describe an IC)



IC Card IC Name: Help

Description: Perform appropriate action for help

Interaction Pattern:

Quiet State | By Myself no Interaction | By Myself with Interaction | By Others no Interaction | By Others with Interaction | Mixed

My Task: Call emergency responder

Time Critical Condition: Needs to be done immediately

Name of Other IC: Emergency Responder

Message to Other IC: Patient needs help!

Other IC's Task: Call or visit the paitent

Card 1 of 1 (If necessary please use several IC cards to describe an IC)

Card two obviously includes a lot of work – the system must use the similarity metrics GESTURECODE to compare the given gesture with ones in the gesture database, thereby getting the correct action (and thus, command) that needs to be issued by the system. This involves a lot of calculations and working together by different components of the system.

These cards can be arranged in a hierarchy, as so: