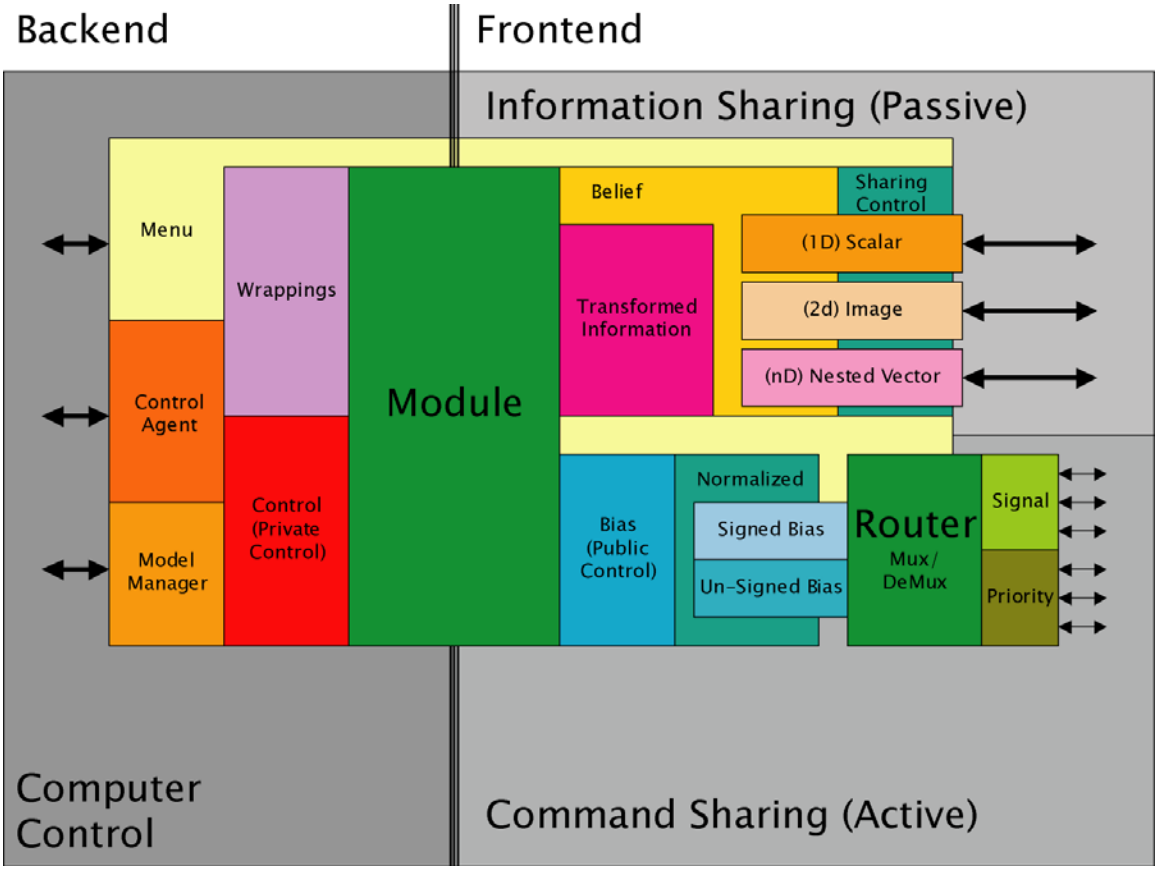
iRoom Expert Module Standard

The iRoom expert module standard (IEMS) is being developed to accomplish several objectives. The first and most notable is it should be developed to allow as effortless as possible sharing of resources authored by different individuals as part of the iRoom development project. Additionally, it should set a standard for information among experts that forces the authors of modules to think about the extensibility of the information which they can provide.

IEMS will be an attempt to create a standard which is flexible enough to allow for a variety of different kinds of experts, but will require a minimum standardization such that others can connect modules with as little effort as possible. As such what will be defined has nothing to do with the internal workings of each module. As such they will be for all intents and purposes black boxes. What will be defined is the resources shared. That is, the computation that goes into creating an image for instance is less important than that the image result is what it says it is. An example might be that a saliency map if shared can be created in any way the author sees fit so long as what comes out meets the definition of a saliency map. Eccentricities of that saliency map will be communicated if necessary via wrapping resource sharing.

IEMS defines three types of information to be utilized by each module. The first is the back end. This is information that is primarily private. These are control parameters and

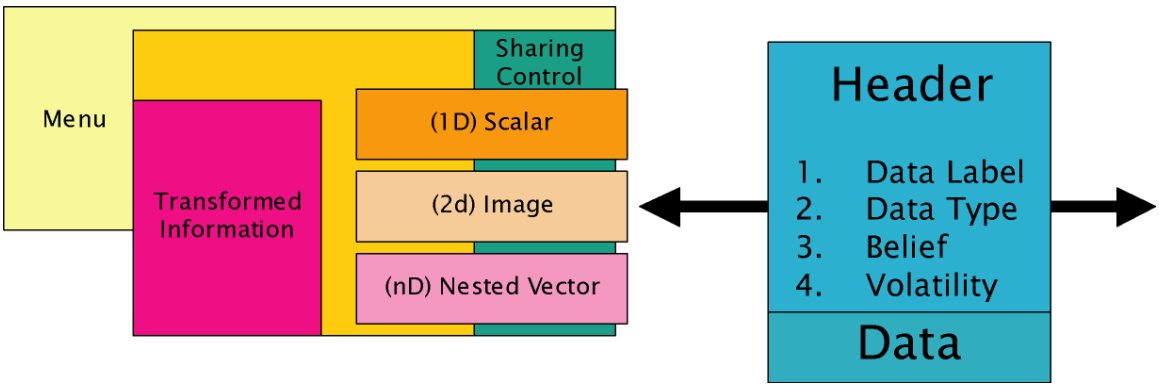


T Nathan Mundhenk

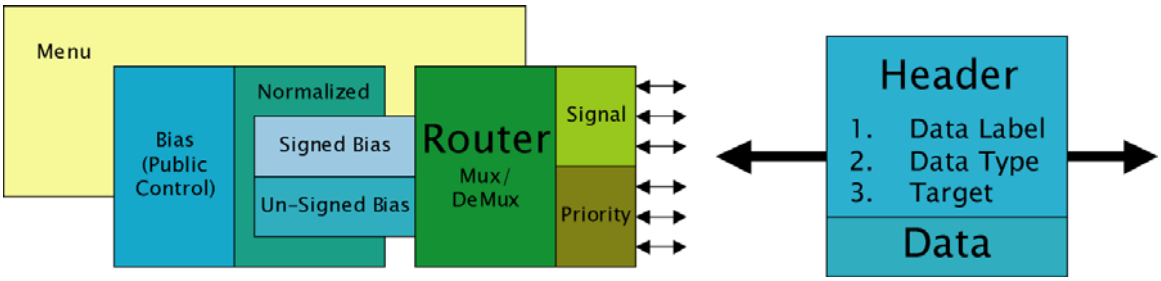
Digitally signed by T Nathan Mundhenk
DN: CN = T Nathan Mundhenk, C = US, O = University of Southern California, OU = USC
Reason: I am the author of this document
Date: 2005.04.27 14:42:00 -07'00'

initial variables need only be understood by the author. For instance, initial sizing of arrays may be passes via the backend.

The other two types of information passed by the module are the frontend. These are broken into passive and active components. Passive information can be thought of as information sharing and does not necessarily compel another module to act. Active information on the other hand is designed to attempt to change another modules behavior overtly. The difference between passive and active information can be thought of like this; Passive information might be “Hey did you hear the Romans are invading” while active information might be like “Lets get out of here”. The first only shares information, what the agent does with that information is up to the agent. In this case, the agent might like the Romans, so it decides to stay. The later example would be if your friend trys to persuade you to leave. As such active and passive components can interact or be separate.



Passive information should be designed to maximize logical information in a manner that is coherent between modules. It is suggested that it be constrained to scalars, images and STL vectors. Additionally, each passive communication should be accompanied by a belief tag. This is the degree to which the module passing the information expresses confidence in the information. Since each module is supposed to be an expert, it must also express the confidence in its results.



Active information is in the form of biases. A routing/mixing layer will handle biases as they come in from multiple modules. A bias should be related to either internal states of a module that are known or passive information that has been passed. As such one module may simply compel another module to alter it’s behavior. The other module however, may choose not to be compelled. From a communication standpoint the receiving module must communicate this by either creating and echo about what it will do or the sending module must observe the receiving modules behavior and attempt to bias the receiving

module even more if need be. This means that any module may compel another module to act if it is connected and it acts with a strong enough bias. Bias information should probably be normalized to be either between –1 an 1 or 0 and 1. This will allow authors to know for sure the maximum and minimum biases that can be transmitted without having to spend time to look it up.

Modules will integrate information and resource sharing via a wrappings protocol. Each module will have a *menu* where it will list what resources it has and needs in as simple and efficient a manner as possible. This will list for instance all the biases that are available as well as the information shared passively. This will allow either a software agent or a human operator to connect modules as easily as possible.