Graphical Interface for Creating Compounded Components in CURE

Patric Jensfelt and John Folkesson

We are developing a new software system for robotics at CAS called CURE which is short for CAS Unified Robot Environment. At the lowest level in CURE the components all have well defined inputs and outputs. These components can be connected to form more complex components. This is done by connected inputs and outputs of the different components. There are different types of connections

• push
  Pushes information from the output side to the input side. This is the standard way of connecting components. If you put a number of components in series with push connections in between a single input in one end will make all components process the input sequentially, passing the result on to the next level.

• pull
  With type of connection the input side can ask for information when needed from the connected output.

Some examples of basic components are:

• LineFilter
  Takes a scan from a laser range scanner and extracts lines from it.

• OdomModel
  Takes the raw odometric information from wheel encoders on a robot and calculates cumulative position and incremental positions including uncertainties

Figure 1 shows how a number of components are connected to implement a localization system that takes laser scans and odometric data (wheel encoder data) as input and produces a position estimate as output.

These compounded components are currently created by writing code that performs the connections between the individual components. This project would look at ways to perform these connects using a graphical user interface instead. The code for the higher level component could then be automatically generated from the graphical model (compare figure).

The task of the project would be to design such a graphical user interface, generate the code so that the components can be used in programs and come
Figure 1: Example of how filters are compounded into more complex filters.

up with a way to specify the individual components such that it is easy to add new components to the system. To limit the scope of the project there is no requirement that there be any means of reading in code and automatically generating a graphical model for it. All modifications will be done in the graphical model. It must however be possible to use the generated components in other compounded components at a higher level.