Client-server or master-slave coordination occurs when a *master* assigns a *slave* a job to perform. This strategy necessarily requires a *superior* relationship between the slave and master. The *contract-net* strategy is when an agent announces a task, requesting bids to perform the task from other agents. It then evaluates the bids and awards the contract to the winning agent(s). This strategy does not rely on any organisational relationship between the participating agents. *Limited contract-net* is similar to the contract-net strategy except that the announcer selects/restricts the agents to whom bids are broadcast.

The coordination protocols listed above can be single round or multiple round. A single round behaviour between two agents involves some agent receiving some proposal (for example) and accepting it immediately. Alternatively, the agent may initiate another round of the cycle in which case a multiple round behaviour emerges. The key point here is that the *terminating condition* on the agent that initiates the interaction must be met before the interaction terminates.

Typically, most other agent applications have agents with one or at most two fixed strategies. CABS agents can have many more as illustrated by the list above. Hence for example, an agent may use a first coordination protocol (say the English Auction) to sell some antique while *simultaneously* using a second protocol (e.g. Dutch Auction) for selling flowers, whilst at the same time using a third protocol (the contract net) for some other tasks distribution.

If a user requires a coordination strategy which is not provided by the CABS system, the Coordination Editor 330 can be used to define a new coordination graph to add to the agent coordination graph database 255. Using the UCP as starting point, new co-ordination graphs can be created for instance by adding a new sub-graph which introduces new nodes between some of the existing nodes in one of the predefined graphs, and/or by modifying one or more of the arcs in a predefined graph. Alternatively, a completely new graph can be created by following the process described earlier under "2.4.1 Coordination Software Module: Defining new coordination graphs". The key advantage of this novel approach is that it maintains a simple core (the UCP) while facilitating considerably more complex coordination techniques to be developed and installed. New coordination graphs can be assembled from primitives provided, based on the