A particularly important feature of the CABS agent is that its co-ordination engine 210 can implement more than one co-ordination process, and therefore more than one coordination protocol, *simultaneously*. This is done effectively by multiplexing execution of the co-ordination graphs held in the co-ordination graph database 255 by an agent. The engine 210 deals with the first node of each co-ordination graph 255 it has been loaded with, then steps on sequentially to the second nodes of all the graphs, etc. Thus it effectively steps through the co-ordination graphs 255 in parallel.

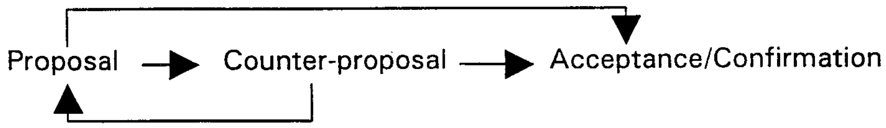
CABS thus provides in the agent shell 300 a co-ordination engine and reasoning system 210 for which the functionality is determined by selection from a set of co-ordination graphs 310 during agent build. Once each agent is in use, the co-ordination graphs are used by the co-ordination engine and reasoning system 210 to run specified co-ordination process steps and protocols.

The repository of process steps identified by the labels is not necessarily itself held in each agent. It may simply be accessed by the agent at runtime, in accordance with its co-ordination graph(s).

(In the following, the co-ordination engine and reasoning system 210 is also referred to as the Coordination Software Module 210.)

The overall architecture of the Coordination Software Module 210 is one of a Turing state machine. It takes as inputs various state variable values, parameters, goals, exceptions and constraints, and it outputs decisions.

In a multi-agent system, when two or more agents go through a co-ordination process, using their respective Coordination Software Modules 210, the overall process functionality can generally be represented by a "universal co-ordination protocol (UCP)" as follows:



UCP