the third propose message was not sent out would be (once more) that no agent is listed in agent C's aquaintance database capable of performing the third job.

The overall hypothesis that the failure lies in agent C can be verified by, for example, using the micro tool for viewing the internals of agent C. For instance, it might be noted from the trace of its co-ordination graph that agent C could not plan or co-ordinate the activities the activities involved in J2, thus confirming the hypothesis.

Regarding the hypotheses (i) and (ii), the control tool can be used to remotely review the task specification database of agent C. If that is correct, (i) is eliminated. To confirm hypothesis (ii), the control tool can be used to directly review the aquaintance database of agent C, checking all specifications relating to J21, J22 and J23 (by browsing through the aquaintance database of agent C using a task browser window of the control tool).

Alternatively, if the aquaintance database is large, it is possible to use the society tool to review the messages sent out by agent C. It might be discovered that agent C sent out messages to agents F and G to perform jobs J22 and J23 respectively. Now it is reasonably certain that the failure to perform job J was due to a specification error in the aquaintance database of agent C, regarding J 21 (the goal) and agent E. This can be confirmed by using the control tool to review and modify the aquaintance database of agent C.

It should be noted that the suite of tools described above cannot detect every possible cause of failure in a multiagent system it is used to monitor but it makes the debugging task immeasurably easier.

Although described in relation to the particular CABS multiagent system described above, one or more of the debugging tools could easily be used with other multiagent systems, based on different architectures, agents, relationships and languages. It would simply be necessary that the agents of the debugging tool(s) and of the multiagent system being debugged were able to understand messages and other information, such as task specifications, in the same way so that the debugging tool(s) could understand what the messages concerned and could review and amend the other information.

SUBSTITUTE SHEET (rule 26)