

Figure 2): external messages from other agents into the Mailbox 200 (e.g requests for a service), external events initiated from the Execution Monitor 240 monitoring external systems (e.g. from various sensors) and external events initiated from changes to the Resource Database 225. For example, if there is an event which
5 changes the state of the agent, such as loss of a resource, it will need to update its records accordingly.

A change in state may initiate a sequence of activities within and/or outside the particular agent. For example, losing a resource (e.g. through failure) which is required to provide some service would require the Planner & Scheduler
10 module 220 to attempt to secure another resource which may be able to do the same job. If it succeeds, all activities as a result of the loss of the resource can be contained within the agent. However, if the Planning and Scheduling module 220 cannot locate another local resource, the Coordination Engine and Reasoning
15 System 210 will be called upon to attempt either to secure the resource from some other agent or delegate/contract the task which required that resource to another agent. In both cases, the Coordination Engine and Reasoning System 210 will request the Mailbox 200 via the Message Handler 210 to construct a message and
despatch it to selected other agents. In this way, coordination of activities with other agents is realised.

20 Further details of the components of the agent structure which support the above mechanism are given below.

2.1 Mailbox 200

25 The mailbox 200 is implemented as a multi-threaded module with inter-agent communication via TCP/IP sockets. One thread of execution, the server thread, continuously listens for and accepts incoming TCP connection requests from other agents, receives any incoming messages from those agents and puts the received messages into an in-tray (a queue). A second thread, the client
30 thread, opens TCP connections with other agents to deliver messages. Messages to be delivered are retrieved from an out-tray (queue). When other modules of the agent request a message to be delivered to another agent, those messages are placed on the out-tray of the mailbox to be later picked up by the client-thread. The message language used in the current implementation is the KQML agent