

(iii) a co-ordination engine and reasoning system 210 which takes decisions concerning the goals the agent should be pursuing, how said goals should be pursued, when to abandon them etc., and how to co-ordinate the agent's activities with respect to other CABS agents in the system. The co-ordination engine and reasoning system contains both an engine and a database of coordination processes 255;

(iv) an acquaintance model 215 which describes the agent's knowledge about the capabilities of other agents in the system;

(v) a planner and scheduler 220 which plans and schedules the tasks the agent is controlling, monitoring or managing based on decisions taken by the co-ordination engine and reasoning system 210 and the resources and tasks available to be controlled, monitored and/or managed by the agent;

(vi) a resource database 225 containing logical descriptions of the resources currently available to the agent; and providing an interface between the database and external systems such that the database can query external systems about the availability of resources and inform external systems when resources are no longer needed by the agent, and external systems can on their own initiative add, delete or modify resource items in the database, thus initiating changes in the agent's behaviour;

(vii) a task database 230 which provides logical descriptions of tasks available for control, monitoring and/or management by the agent; and

(viii) an execution monitor 235 which starts, stops, and monitors external systems tasks scheduled for execution or termination by the planner and scheduler, and which informs the co-ordination engine and reasoning system 210 of successful and exceptional terminating conditions to the tasks it is monitoring.

When the agent is built, the developer uses various CABS-provided editors to provide descriptions required for various modules of the agent architecture including the coordination and reasoning engine 210, the acquaintance model 215, the resource database 225 and the task database 230.

In use, the agent is driven by events which cause the agent's state to change. The agent will run an event model provided by the component library of the system to monitor these internal changes, making them visible to a programmer via an API. There are three possible external event sources (see