

the debugging effort. For easy visualisation, the different states of a job can be colour-coded in the graph.

Figure 7 as a whole shows a task report 700 for job J1 owned by Agent A. The subpart J111 (Agent E) has failed, while J12 (Agent C) is running and J13 (Agent D) is completed. J11 (Agent B) and J1 (Agent A) are waiting, but because of the failure to achieve J111 will both fail unless action is taken to achieve J111 in some other way.

The tool also provides the user with facilities for collapsing/expanding sections of the graph and hiding/showing nodes on the graph – this is important in dealing with a very large graph since it allows a user to focus-in on the regions of interest, reducing too much detail which might hinder the debugging effort.

The algorithm for ensuring the correct integration of task descriptions into graphs relies on two facts: first, the agent on which the goal is initiated assigns a system-wide unique identifier to the goal that is propagated down the task decomposition hierarchy of the goal. Second, whenever an agent decomposes a task into subtasks, it creates unique identifiers for the subtasks and records the identifiers with the parent task (this happens whether or not other agents perform the subtasks). Thus, using the root goal identifier and parent-child links, the report tool can graph the global task decomposition across the society. (In fact, the same mechanism is used in the society tool to colour-code messages by goals.) The same mechanism also allows the display of joint graphs, wherein two task decomposition graphs are linked by one or more tasks. This happens when side effects of a task in one goal decomposition graph are utilised in another.

Similarly to the society tool, the report tool also supports online logging of report messages to a database, and then subsequent off-line replay.

### 5.3 The Micro Tool

Referring to Figure 14, the micro tool allows a user to select a single agent, Agent C as shown, and review its processing by looking at its current data and at the messages being sent and received by different components of the agent. For instance, it might look at the last ten messages for a selected component. Looking at threads in the mailbox, it might review messages under construction to go out, or messages recently received. It can request to see: