provide a description of the summary tasks in terms of its component tasks.

Tasks consume certain resources to produce their products, with each executing

5  task lasting for a finite time interval. The CABS platform provides a *Variables Editor* 355 using a frame-based object-attribute-value representation for defining the consumed and produced items.

The constraints between and within the consumed and produced items serve to delimit each task to specific well-defined cases. For instance, a task may

10  call on a particular resource but there may be a constraint on the resource for that particular task, such as processing capacity or equipment type. A *Constraints Editor* 370 is also provided to describe constraints between the variables to ensure that tasks use realistic variables and that each task output equals the input to the next stage. This can be extended to setting inequalities, for example saying what

15  an output should <u>not</u> be, for example integer/non-integer.

In addition, a *Summary Task (Plan) Editor* 365 is provided for listing component tasks of summary tasks. This editor allows summary tasks to comprise (i) simple actions (ii) multiple optionally parallel actions (iii) multiple (mandatory) parallel actions (iv) guarded actions (selections), and (v) iterative

20  actions. Furthermore, these actions can be linked as a network, with a facility to constrain the variables at the endpoints of each link.

Below is an example of a task description created using the task description editor 335. The task describes the fact that to create a link from A (?Link-141.from) to B (?Link-141.to) a link is needed from A (?Link-131.from) to

25  some location X (?Link-131.to) and another link from X (?Link-136.from) to B (?Link-136.to).

The consumed_facts of the task list the resources required in performing the last, i.e. the links from A to X and from X to B. The produced_facts lists the resource which will be produced once the task is performed, i.e. the link from A to

30  B. The constraints specify relationships between the consumed and produced facts. For example, the constraint "?Link-141.from = ?Link131.from" specifies the condition that the source of the produced link should be the same as the source of one of the required links. These and similar constraints will be enforced