go to next stage (S_{11} - S_{12});

- if no confirmation received

  reject and terminate (S_{11} - S_E);

Stage Thirteen

Send confirmation (S_{12} - S_{13});

Stage Fourteen

Firmly reserve the resources and terminate (S_{13} - S_{14}, S_{14} - S_E);

2.3.3 Defining New Coordination Mechanisms

The design of the coordination machine allows virtually infinite extension to the various UCP-compliant coordination graphs and even to non-UCP-compliant graphs. This is achieved by the fact that the coordination engine is a generic and domain-independent function which interprets any graph description of the form described earlier. By giving different coordination graphs to the engine, different coordination behaviour will be performed. Moreover, by altering the implemented functions (i.e. programs) identified by the node and arc labels of a graph, then on executing that graph the engine will behave differently even if the set of given labels remains unchanged.

It has been realised that the UCP and its derivative, the 14-stages framework, subsume many existing coordination mechanisms. It has also been realised that these mechanisms differ only in certain minor, but subtle, aspects. After careful analysis, it was found that by differentiating the UCP in 14 stages, most mechanisms have many of the stages in common and only differ in a small number of stages. This makes it possible to implement a large number of different mechanisms by developing only a few new functions and many of the common functions can be reused. In addition, specifying mechanisms using a unified underlying framework makes it possible to solve a problem using a mixture of different mechanisms.

Given a coordination behaviour, in order to define a new UCP-compliant coordination graph, the following steps should suffice: