communication language [Tim Finin, Yannis Labrou & James Mayfield (1997),
KQML as an Agent Communication Language, in Bradshaw, J (Ed.), Software
Agents, Cambridge Mass: MIT Press, Chapter 14, pages 291-316]

5   The mailbox technology is relatively standard and could have been
implemented using alternative communication protocol other than TCP/IP such as
electronic mail and HyperText Transfer Protocol (HTTP).

2.2 Message Handler 205

10   The message handler 205 continually polls the in-tray of the mailbox 200
for new incoming messages, which it dispatches to other relevant components of
the agent for detailed processing. This module is based on standard technology,
comprising a continuously running thread which polls the mailbox 200, and has
access to handles of all the major components of the agent for dispatching
15 messages to them.

      In CABS, the format of KQML messages is as follows:

      KQML [:sender :receivers :content]

20   Agent messages, including inter-agent communications as usually used in
co-ordination in CABS, use KQML performatives, and the details of the
communications are usually contained in the KQML content field. A CABS agent
that puts forward a change in state to other agents (which may represent a
proposal, counter-proposal, or acceptance in the CABS co-ordination protocol) uses
25 the KQML performative *"achieve"*. Recipient agents reply by using the KQML
performative *"reply"* when they counter-propose and *"accept"* when they accept a
proposal.

2.3 Co-ordination Engine And Reasoning System 210

30
      Referring to Figures 2 and 3, the co-ordination engine and reasoning
system 210 and the planner and scheduler 220 are both unique in a CABS agent
and both are now described in detail. Further reference is made to these below, in
describing use of the CABS platform to design an agent system, under the