

19

14. The system of claim 13, wherein the collaborative client further comprises:

means for instantiating a collaborative team; and
team descriptor logic means for ordering of events posted to a team and dispatching of received events.

15. The system of claim 14, further comprising:
an optimistic event execution model; and

team descriptor logic means for ordering events through the server and dispatching locally generated events immediately.

16. The system of claim 15, said conflict detection means further comprising means for detecting an out-of-order event based on an event notification received from the server.

17. The system of claim 1, wherein the events are the units of communication in the collaborative client, further comprising:

Late Event Modification (LEM) means for batching and modifying events after posting them to the shared set.

18. The system of claim 1, said conflict detection means further comprising means for detecting conflicts due to incorrectly ordered updates to the shared object and informing the application.

19. In a distributed collaborative application wherein one or more distributable components may be executed independently and communicate with other components, each component including one or more sets of replicated shared objects, a method for updating replicated state, comprising the steps of:

a client communicating updates to and receiving updates to the state of a shared object;

detecting diverging state among the shared object sets; and

cloning and reinitializing the state among the shared object sets, in response to said detecting.

20. The method of claim 19, said detecting step further comprising the steps of:

detecting conflicts due to incorrectly ordered updates to the shared object; and

informing the application, in response to said detecting step.

21. The method of claim 19, further comprising the step of:

marshalling and unmarshaling clones of one or more shared objects in the shared object sets.

22. The method of claim 19, wherein said cloning and reinitializing step further comprises the steps of:

making a copy of a subset of objects in the set and preserving pointers from inside the original subsets to objects outside the cloned subset;

mapping references between objects inside the original subset into the cloned subset;

exporting the cloned subset to other clients; and

re-initializing the subset of objects based on the cloned subset, in response to said exporting step.

23. The method of claim 19, further comprising the steps of:

marshalling and propagating context-sensitive state to one or more new or updated shared objects in the shared object sets.

24. The method of claim 19, wherein the system is a client-server system, further comprising the steps of:

said receiving step comprising receiving a shared event from the server; and

detecting an out-of-order event based on the shared event received from the server.

20

25. The method of claim 19, wherein the system is a client-server system, each client further comprises a collaborative client middleware, communicatively coupled to the application and to the network.

26. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for a method for updating replicated state in a distributed collaborative application wherein one or more distributable components may be executed independently and communicate with other components, each component including one or more sets of replicated shared objects, said method comprising the steps of:

a client communicating updates to and receiving updates to the state of a shared object;

detecting diverging state among the shared object sets; and

cloning and reinitializing the state among the shared object sets, in response to said detecting.

27. The program storage device of claim 26, wherein each client comprises a collaborative client middleware, communicatively coupled to the application and to a network.

28. The program storage device of claim 26, wherein the system is a client-server system, further comprising the steps of dynamically creating, updating and destroying a collaborative team, wherein each team has a unique name space in a collaborative activity on the server.

29. The program storage device of claim 28, further comprising the step of defining and implementing a team behavior based on a team policy.

30. The program storage device of claim 26, further comprising the steps of:

said detecting diverging state comprising the step of detecting conflicts due to incorrectly ordered updates to a shared object; and

informing the application, in response to said detecting step.

31. The program storage device of claim 26, further comprising the step of:

marshalling and propagating context-sensitive state to one or more new or updated shared objects in the shared object sets.

32. The program storage device of claim 26, further comprising the step of:

marshalling and unmarshaling clones of one or more shared objects in the shared object sets.

33. The program storage device of claim 26, wherein said cloning and reinitializing step further comprises the steps of:

making a copy of a subset of objects in the set and preserving pointers from inside the original subsets to objects outside the cloned subset;

mapping references between objects inside the original subset into the cloned subset;

exporting the cloned subset to other clients; and

re-initializing the subset of objects based on the cloned subset, in response to said exporting step.

34. The program storage device of claim 26, wherein the events are the units of communication in the collaborative client, the method further comprising the steps of:

posting an event to the shared object set;

batching and modifying events after posting them to the shared object set; and

marshalling and propagating context-sensitive state to one or more new or updated shared objects in the shared object sets.