knowledge of the physical address of the second object on the second base; and

at least one runtime system connected to the first base and the second base, the at least one runtime system facilitating migration of agents and objects from at least the first base to at least the second base.

2. The distributed software system of claim 1, wherein each agent further comprises at least one subagent, each subagent residing on one base and comprising:

an object memory which stores objects in the subagent;
a task memory which stores task frames in the subagent;
program code for the agent to which the subagent belongs;
a subagent control storage comprising:

an agent identifier indicating the agent to which the subagent belongs;
an object table having a mapping which maps symbolic references of objects to corresponding physical addresses of said objects in the object memory;
a task stack which stores a plurality of task thread pointers in the task memory;

and wherein the at least one runtime system further comprises:

an agent manager for each base managing a plurality of subagent control storages of subagents residing on the corresponding base;
an object manager for each base managing a plurality of object memories for a plurality of subagents residing on the corresponding base;
an object serializer for each base serializing objects for transmitting the objects across the network to at least one base other than the corresponding base;
a task executer for each base reading program code, creating task stacks in task memories, and executing the program code;
a task serializer for each base serializing task stacks for transmitting the stacks across the network to at least one base other than the corresponding base;
a remote access controller for each base receiving remote object access messages from a task executer on at least one base other than the corresponding base and sending remote object access requests to at least one base other than the corresponding base; and
a communication system coordinating physical communication between the computer

machine and the other computer machines.

3. The distributed software system of claim 2, wherein the program code is stored as class files in the sub-agent.

4. The distributed software system of claim 2, wherein the runtime system further facilitates migration of tasks from the first base to the second base.

5. The distributed software system of claim 1, wherein the first object is a task and the second object is a data object.

6. The distributed software system of claim 1, wherein the at least one agent further comprises a global object space, wherein the global object space includes a mapping of symbolic references of objects within the at least one agent to corresponding physical addresses of said objects, whereby the first object on the first base accesses the second object on the second base without knowledge of the physical address of the second object on the second base by obtaining a symbolic reference to the second object from the first object and obtaining the corresponding physical address of the second object using the mapping of the object space.

7. The distributed software system of claim 6, wherein the object space is implemented using global identifiers for addressing each object.

8. The distributed software system of claim 1, wherein the access by the first object of the second object is a method call specifying at least one of an argument and a return value, wherein a symbolic reference to the at least one argument or return value may be passed to or returned from the called method to identify the at least one argument or return value, and wherein the physical address of the at least one argument or return value need not be passed to or returned from the called method to identify the at least one argument or return value so as to render the method call network transparent.

9. The distributed software system of claim 8, wherein the at least one agent further comprises a global object space, wherein the global object space includes a mapping of symbolic references of objects within the at least one agent to corresponding physical addresses of said objects, whereby the system permits the symbolic reference to the at least one argument or return value of the method call to be passed to or returned from the called method to identify the at least one argument or return value by obtaining a symbolic reference to the second object from the first object and obtaining the corresponding physical address of the second