object using the mapping of the object space.

10. The distributed software system of claim 9, wherein the object space is implemented using global identifiers for addressing each object.

11. The distributed software system of claim 1, wherein:

the first object is a first task residing within the protection domain of the at least one agent and executing within the first base; and

the second object is a second task residing within the protection domain of the at least one agent and executing within the second base, wherein the first task and the second task execute concurrently on the first and second bases, respectively, and within the same protection domain of the at least one agent.

12. The distributed software system of claim 1, wherein each base may provide the local address space and computer resources to a plurality of agents simultaneously.

13. The distributed software system of claim 1, wherein the first base and the second base are located in heterogeneous computer machines.

14. A distributed software system for use with a plurality of computer machines connected as a network, the system comprising:

at least one agent comprising a protection domain, wherein the protection domain of the at least one agent resides on at least two of the plurality of computer machines; and

a plurality of objects contained within the protection domain of the at least one agent, the objects being selectively movable among the at least two computer machines by a programmer of the system, a first object residing on a first of the at least two computer machines and a second object residing on a second of the at least two computer machines, wherein the first object on the first computer machine may access the second object on the second computer machine without knowledge of the physical address of the second object on the second computer machine, and regardless of the selective movement of either the first object or the second object among the first and second computer machines.

15. The distributed software system of claim 14, wherein the first object is a task and the second object is a data object.

16. The distributed software system of claim 14, 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 computer machine accesses the second object on the second computer machine without knowledge of the physical address of the second object on the second computer machine 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.

17. The distributed software system of claim 16, wherein the object space is implemented using global identifiers for addressing each object.

18. The distributed software system of claim 14, 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.

19. The distributed software system of claim 18, 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 object using the mapping of the object space.

20. The distributed software system of claim 19, wherein the object space is implemented using global identifiers for addressing each object.

21. The distributed software system of claim 14, wherein:

the first object is a first task residing within the protection domain of the at least one agent and executing within the first computer machine;