FIG. 4B is a schematic diagram showing weak migration of an agent from a source base to a destination base in a distributed agent system according to the present invention;

FIG. 4C is a schematic diagram showing complete migration of an agent from a source base to a destination base in a distributed agent system according to the present invention;

FIG. 5 is a schematic diagram showing a server handling a request for an agent to migrate to a particular base served by the server in a distributed agent system according to the present invention;

FIG. 6 is a flow diagram showing an exemplary method for implementing the network-centric migration of the present invention;

FIG. 7 is a schematic diagram of an agent running on two separate bases, and illustrating method calls under both an RPC model and an invoker model in a distributed agent system according to the present invention;

FIG. 8A is a schematic diagram showing migration of an object in core library classes in a distributed agent system according to the present invention;

FIG. 8B is a schematic diagram showing object migration in a user-defined class in a distributed agent system according to the present invention;

FIG. 8C is a schematic diagram showing new class object creation in a distributed agent system according to the present invention;

FIG. 9 schematic diagram showing the relationship of runtime systems to the other basic components of a distributed agent system according to the present invention;

FIG. 10 schematic diagram showing subcomponents of bases, subagents and runtime systems of a distributed agent system according to the present invention;

FIGS. 11A-11E are schematic diagrams showing the relationship of runtime systems of the present invention to an exemplary sequence of agent migration in a distributed agent system according to the present invention;

FIGS. 12A and 12B are schematic diagrams showing the relationship of runtime systems of the present invention to an example of partial agent migration in a distributed agent system according to the present invention;

FIGS. 13A and 13B are schematic diagrams showing the relationship of runtime systems of the present invention to an example of whole agent migration in a distributed agent system according to the present invention;

FIGS. 14A and 14B are schematic diagrams showing the relationship of runtime systems of the present invention to an example of object migration in a distributed agent system according to the present invention;

FIGS. 15A and 15B are schematic diagrams showing a first example of remote object access in the context of agent migration in a distributed agent system according to the present invention;

FIGS. 16A and 16B are schematic diagrams showing a second example of remote object access in the context of agent migration in a distributed agent system according to the present invention; and

FIG. 17 illustrates an instance object and a class object for use in a distributed agent system according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0034] Generally speaking, the invention is implemented in an object-oriented language. For example, the invention may be implemented in conjunction with the Java language. To facilitate the description herein, descriptions of language syntax and core semantics are written so as to generally resemble Java, and certain Java terminology is used (e.g. "static methods," "instance fields," "self," etc.) and will be obvious from the context. It should be noted, however, that the invention is not limited to a Java implementation and may be implemented in any suitable object-oriented language.

[0035] To assist the reader in understanding the following description, the following definitions are useful:

"Object": An object is an instance of a class, and represents a primitive storage unit for any kind of information.

"Object space": An object space is a collection of objects and tasks that may freely reference one another.

"Protection domain": A protection domain is a software structure which prevents unauthorized access to the data within it by computation executing outside the protection domain.

Basic Structure

[0036] Reference is now made to FIG. 1, which depicts basic architectural components of the unique distributed agent model of the present invention. A plurality of nodes or machines 10, such as machines 10a and 10b, are connected to each other through a communication interface 20 to form a network 25. Machines 10a and 10b may be homogeneous or heterogeneous machines. Each machine includes one or more "bases" 30, such as bases 30a, 30b and 30c. Agents 40, such as agents 40a, 40b and 40c, are also provided, each of which runs on one or more bases 30.

[0037] Each base 30 has a unique identifier and provides a local address space and resources on a machine 10. Each base is preferably implemented as an operating system-level process, such as a UNIX process or Windows process. In the simplest case, a base runs on one processor; however, a base may also run