NCS TIB 02-1

NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 02-1

Network Planning for 21st Century Intelligent Systems and Smart Structures: Advanced Concepts in Telecommunications for National Security and Emergency Preparedness (NS/EP)

February 2002

OFFICE OF THE MANAGER
NATIONAL COMMUNICATIONS SYSTEM
701 SOUTH COURTHOUSE ROAD
ARLINGTON, VIRGINIA 22204

NCS TECHNICAL INFORMATION BULLETIN 02-1

NETWORK PLANNING FOR 21st CENTURY INTELLIGENT SYSTEMS AND SMART STRUCTURES: ADVANCED CONCEPTS IN TELECOMMUNICATIONS FOR NATIONAL SECURITY AND EMERGENCY PREPAREDNESS (NS/EP)

February 2002

PROJECT OFFICER

DALE BARR, JR.
Sr. Electronics Engineer
Technology and Programs Division

APPROVED FOR PUBLICATION:

PETER M. FONASH
Chief, Technology
and Programs Division

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System, is the management of the Federal Telecommunications Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunications Standards Committee identifies, develops, and coordinates proposed Federal Standards which either contribute to the interoperability of functionally similar Federal telecommunications systems or to the achievement of a compatible and efficient interface between computer and telecommunications systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development efforts with appropriate technical committees of the International Organization for Standardization, the International Telecommunication Union-Telecommunications Standardization Sector, and the American National Standards Institute. This Technical Information Bulletin presents an overview of an effort which is contributing to the development of compatible Federal and national standards in the area of national security and emergency preparedness (NS/EP) . It has been prepared to inform interested Federal and industry activities. Any comments, inputs or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

Office of the Manager
National Communications System
Attn: N2
701 S. Court House Road
Arlington, VA 22204—2198

Network Planning for 21st Century Intelligent Systems and Smart Structures: Advanced Concepts in Telecommunications for National Security and Emergency Preparedness (NS/EP)

Office of the Manager

National Communications System

February 2002

Communication Technologies, Inc.

14151 Newbrook Drive, Suite 400

Chantilly, Virginia 20151

703-961-9080 (Voice)

703-961-1330 (Fax)

http://www.comtechnologies.com

Abstract

The National Communications System (NCS) in support of its primary mission is exploring revolutionary communication concepts for the 21st century. These concepts are in the specialized fields of smart materials and structures, advanced network architectures, and intelligent systems. The purpose of this report is to assess the state-of-the-art and state-of-the-practice in these technologies and in intelligent software agents related to these concepts. Further, the ability of the telecommunications infrastructure to support agent-based monitoring, detection, planning, cooperation, and decision-making in support of National Security and Emergency Preparedness (NS/EP) situations was also investigated. Specificity, the report focuses on the evolving network infrastructure, wireless networks and Internet-based networks, to support both intelligent systems and smart structures.

The finding of this report suggest that the NCS should work with its governmental and industrial partners to promote NS/EP requirements in there specifications, designs and implementations of these revolutionary concepts so they can be seamlessly incorporated into legacy and future communications systems. In support of these efforts, NCS should create an environment that fosters the development of proof-of-concept experiments exploring the integration of these technologies in the context of NS/EP situations and scenarios; wherein, integrated intelligent systems incorporating smart structures could combine wireless and Internet technologies to allow devices to communicate over the Internet and Public Switched Telephone Network (PSTN).

Table of Contents

List of Figures

List of Tables

Executive Summary

The National Communications System (NCS) in support of its mission is exploring revolutionary concepts for the 21st century, in the areas of smart materials and structures, advanced network architectures, and intelligent systems. Clearly each area is a specialized field with hundreds of scientists and engineers pursuing research and development in these diverse areas. While each of these areas should be monitored for new results that could impact National Security and Emergency Preparedness (NS/EP), true synergy will only be obtained by integrating concepts from the various fields.

The purpose of this report is to assess the state-of-the-art and state-of-the-practice in the areas of smart sensors and structures, intelligent systems, intelligent software agents, and telecommunications infrastructure to support agent-based monitoring, detection, planning, cooperation, and decision-making in support of NS/EP situations. This report focuses on the evolving network infrastructure, wireless networks and Internet-based networks [1], to support both intelligent systems and smart structures.

Based on research conducted to date, future integrated intelligent systems incorporating smart structures will have the following properties:

- They will combine wireless and Internet technologies to allow devices to communicate over the Internet and PSTN networks.
- The Internet will evolve into the extended Internet, in which devices, such as cell phones, personal digital assistants, and sensors, will download programs to enable them to communicate with other devices.
- There will be billions of these devices that will result in the development of new paradigms and architectures to manage the complexity of simultaneous access to the Internet.
- A promising new field is that of agent-based systems, in which autonomous intelligent agents (proxies for people and devices) will communicate and collaborate with each other to achieve goals.
- Agent communication protocols and languages will allow autonomous agents to discover, to negotiate, to broker, and to subscribe to services. The processing paradigm will be multicast and event-based, and agents will be notified when events to which they have subscribed occur.
- The emerging consensus is that the intelligent network will provide value-added services, such as restrictive call reception, that can be configured and managed by end users.
- The component-based technologies will allow for flexible applications that can be configured from reusable components, are capable of evolution, and can be more easily integrated with legacy systems.

- Wrapper technology will allow integration with legacy systems.
- Extensible Mark-up Language (XML) will become the common document interchange language for devices and applications.
- Smart structures will be built using advanced material and control system technology so as to react and compensate for adverse situations such as earthquakes and hurricanes.

The following is a synopsis of the major areas discussed in this report:

Extended Internet. The extended Internet will help support the concept of smart sensors, smart structures, and agent-based systems. In the future, the number of Internet-aware devices – devices that can connect to the Internet and speak the Internet Protocol – will increase and their computing potential will allow them to perform intelligent tasks. The advances in next-generation internet-based devices and sensors will allow them to communicate with intelligent agents to work cooperatively in solving NS/EP problems, for example, the detection of a bridge collapse and subsequent alternate evacuation route planning. Within the extended Internet, these devices are likely to be organized into complex goal-driven systems that can be used for commerce, education, man-machine interactions.

Transducers. Transducers, which can be either sensors or actuators, are used in manufacturing, industrial control, automotive, aerospace, building, biomedicine, etc. Transducer manufacturers seek to build low-cost, networked smart transducers. In order to interface smart transducers with diverse networks with their associated protocols, the Technical Committee on Sensor Technology of the Institute of Electrical and Electronics Engineers (IEEE) Instrumentation and Measurement Society has sponsored a series of projects, called IEEE P1451 Smart Transducer Interface for Sensors and Actuators.

Sensor Web. National Aeronautics and Space Administration (NASA), in conjunction with Cal Tech's Jet Propulsion Lab (JPL) is developing a web of sensors, called Sensor Web, which will be used to hunt for life on Mars. The Sensor Web consists of a collection of sensors, which measure temperature, light-level, oxygen level, humidity and soil moisture; the pods can be deployed over large areas. The smart sensors consume very little power, are able to communicate among themselves intelligently using advanced protocols, and connect to Internet-based controllers that can process the information.

Smart Structures. The term "smart structures" refers to smart materials, intelligent materials, active materials, adaptive materials as well as actuators and sensors. The goal for smart structures and materials research is to incorporate them into existing materials, or augment existing structures with smart materials so as to control their behavior.

Intelligent Systems. Intelligent systems originally emerged from the artificial intelligence community. However, in practice these early systems were often monolithic rule-based systems, and often became difficult to understand and maintain. More recently, a much more flexible, distributed and adaptable paradigm has emerged for intelligent systems, namely that of intelligent agent-based systems. An intelligent agent-based system is a highly distributed system, in which agents are active concurrent objects that act on behalf of users. Usually, several agents participate in the problem solving activity, communicating with each other. This leads to a more distributed and scaleable environment. Agents can be categorized in different ways, based on their mobility, based on their intelligence, or based on the roles they play in an agent-based system.

Agent-based systems. Intelligent agents communicate with each other in many different ways, either directly or indirectly. For intelligent agent-based systems, the most flexible forms of communication are those that decouple agents from each other, in particular brokered communication and subscription / notification. With these approaches, agents can discover services provided by other agents, register the services they provide, and subscribe to services provided by other agents. In multi-agent systems, it is necessary to allow software agents to negotiate with each other so they can cooperatively make decisions.

Component technology. Component technology provides an environment for agent based systems to evolve more easily and to be more easily integrated with legacy systems, an important consideration for NS/EP systems. An example of this technology is wrappers. Wrappers are software interface programs used to integrate legacy applications into an agent-based system. XML is another example of this technology that allows different agent-based systems to inter-operate through exchange of data and text.

Intelligent agent-based systems for telecommunications applications and services. An intelligent agent-based system is a highly distributed system, in which agents are active concurrent objects that act on behalf of users. Agents are intermediaries between clients, (which are typically user interface objects) and servers (which are typically entity objects) that store persistent information. Usually, several agents participate in the problem solving activity and communicate with each other. This leads to a more distributed and scaleable environment. For telecommunications applications, software agents must have specific capabilities in the areas of agent management, security, personalization, and mobility. Different types of agents are needed – manager agents for administration of the run-time platform, stationary agents as service providers, and mobile agents, which migrate from node to node in a distributed environment, for mobility support and service provisioning.

Summary and Conclusions. There is an opportunity for the integration of smart structures and smart sensors with intelligent agent-based information systems to provide a proactive role in Telecommunications Applications for NS/EP. Smart structures and smart sensors could be used for continuous monitoring of civil structures, such as bridges, roads, dams, and buildings for early detection of emergency situations such as earthquakes, flooding, and hurricanes. Sensors in the atmosphere and at sea can be used for early detection of unusual weather situations such as hurricanes. The sensors can use wireless communications to communicate with local transmitters (for example on buoys at sea or weather balloons) or by satellite communications to the Internet, or by telephone through a gateway into the PSTN. Intelligent agents would receive sensor data from multiple locations from where trends can be determined, for example locality of emergency, speed of movement, and proactively send alarms to interested parties.

Future architectures will enhance the NS/EP feature functionality, capabilities and responsiveness by using families of intelligent agents coupled with smart sensors and smart structures to provide active, intelligent, reasoning systems that 1) respond to stimuli, 2) can react to extraordinary NS/EP situations, 3) are capable of planning, cooperating and taking timely actions, and 4) can learn and adapt over time.

The NCS should work with its governmental and industrial partners to promote NS/EP requirements in the specification, design and implementation of these integrated intelligent systems. NCS should endeavor to work with partners in fostering a testbed to create proof-of-concept experiments to explore the integration of these technologies in the context of NS/EP situations and scenarios.

1 Introduction

DoD has defined six Strategic Research Objectives that focus on multidisciplinary research themes to achieve technology enablement in 10–15 years, with a high potential payoff in numerous Army applications (DoD web page)[1]. Two of these six Strategic Research Objectives are:

- Smart Structures. The objective is to "demonstrate advanced capabilities for modeling, predicting, controlling, and optimizing the dynamic response of complex, multi-element, deformable structures used in land, sea, and aerospace vehicles and systems". Smart structures include smart sensors and smart actuators.
- Intelligent Systems**.** The objective is to "enable the development of advanced systems able to sense, analyze, learn, adapt, and function effectively in changing or hostile environments until completing assigned missions or functions".

The purpose of this report is to assess the state-of-the-art and state-of-the-practice in the areas of smart sensors and structures, intelligent systems, intelligent software agents, and telecommunications infrastructure to support agent-based monitoring, detection, planning, cooperation, and decision-making in support of NS/EP situations. This report investigates the technical and analytical issues in the development of networked advanced systems that use these smart sensors and actuators in adaptive and dynamic distributed real-time control systems.

This report focuses on the evolving network infrastructure, including the PSTN, wireless networks and Internet-based networks [1], to support both intelligent systems and smart structures. The goal is enhance the NS/EP feature functionality, capabilities and responsiveness by using families of intelligent agents coupled with smart sensors and smart structures to provide active, intelligent, reasoning systems that:

- Respond to stimuli,
- React pro-actively to NS/EP situations,
- Are capable of planning, cooperating and taking timely actions, and
- Can learn and adapt over time.

This report is organized as follows: Section 2 reviews NS/EP requirements. Section 3 discusses the emerging distributed Internet-based communications infrastructure and presents a review of standards activities for Internet-based sensors as well as a NASA-JPL-sponsored project in sensor web. Section 4 reviews the area of smart structures, micromachines and their possible use for NS/EP. Section 5 discusses intelligent agent-based systems in general, while Section 6 focuses on their use in telecommunications. Section 7 presents NS/EP scenarios and comments on the use of smart structures combined with agent-based intelligent systems to support NS/EP situations. Sections 8 and 9 present conclusions and recommendations, respectively.

---

[1] See http://www.sarda.army.mil/sard-zt/ASTMP98/vol_i/sec5/sec5b8.htm

2 National Security and Emergency Preparedness Requirements

The NS/EP requirements fall into the following areas [2] [3], as described in the Convergence Task Force Report [2]:

- Enhanced Priority Treatment. Voice and data services supporting NS/EP missions should be provided preferential treatment over other traffic.
- Secure Networks. These services ensure the availability and survivability of the network, prevent corruption of or unauthorized access to the data, and provide for expanded encryption techniques and user authentication.
- Restorability. Should a service disruption occur, voice and data services must be capable of being reprovisioned, repaired, or restored to required service levels on a priority basis.
- International Connectivity. Voice and data services must provide access to and egress from international carriers.
- Interoperability. Voice and data services must interconnect and interoperate with other government or private facilities, systems, and networks.
- Mobility. The ability of voice and data infrastructure to support transportable, redeployable, or fully mobile voice and data communications (i.e., Personal Communications Service (PCS), cellular, satellite, High Frequency (HF) radio).
- Nationwide Coverage. Voice and data services must be readily available to support the National security leadership and inter- and intra- agency emergency operations, wherever they are located.
- Survivability. Voice and data services must be robust to support surviving users under a broad range of circumstances, from the widespread damage of a natural or manmade disaster up to and including nuclear war.
- Voice Band Service. The service must provide voice band service in support of presidential communications.
- Scaleable Bandwidth. NS/EP users must be able to manage the capacity of the communications services to support variable bandwidth requirements.
- Addressability. The ability to easily route voice and data traffic too NS/EP users regardless of user location or deployment status. Means by which this may be accomplished include “follow me” or functional numbering, call forwarding, and functional directories.
- Affordability. The service must leverage new public network (PN) capabilities to minimize cost. Means by which this may be accomplished favor the use of commercial off-the-shelf (COTS) technologies and services and existing infrastructure.

- Reliability. The capability of an information or telecommunications system to perform consistently and precisely according to its specifications and design requirements, and to do so with high confidence.

3 Distributed Internet-based Communications for Intelligent Systems

This section examines the trends in sensor-based, active intelligent networks to support of NS/EP requirements. The advances in next-generation internet-based devices and sensors will allow these sensors to communicate with intelligent agents that work cooperatively in solving NS/EP requirements and problems. The September 2001 special issue of the Communications of the Association of Computing Machinery (CACM) refers to the future Internet as the "Invisible Internet – Always On, Always Available", predicting that by 2006, there will be almost a billion devices interacting worldwide on the Internet [4, 5].

3.1 Forrester Research's X-Internet Concept

Forrester Research has posited a collection of diverse heterogeneous devices that will access the next-generation Internet, the executable Internet, called the X-Internet. The paradigm envisioned by Forrester for the X-Internet represents a vision of an executable network, as compared to a passive collection of Web pages. This re-orientation of the Internet will have wide-reaching implications for the types of devices that connect to it, and the possible future interactions among devices. In the future, the number of Internet-aware devices – devices that can connect to the Internet and speak the Internet Protocol – will increase and their computing potential will allow them to perform intelligent tasks.

To compare the current Internet to the X-Internet, Colony [6] uses the metaphor of someone reading a book to a friend, as compared to a full-fledged conversation. As more and more devices access the Internet, they will have sufficient memory so that programs will be loaded into the device as it negotiates access, has conversations, and executes transactions with a server or another peer device. Thus the X-Internet will execute programs on devices, and exchange very little data. Compare this to the present-day dumb browsers used to navigate the World Wide Web (WWW), also referred to simply as the Web.

The Forrester vision is evolutionary, and starts from the Web of today, goes to the X-Internet of the near future, and then to the extended Internet of 2005 and beyond. While the X-Internet will consist of executable programs, the extended Internet will consist of smart sensors, communicating with smart devices, supported by smart services. Forrester predicts that billions of devices, that can execute real-time business transactions on sensed-data, will reside on the extended Internet.

The extended Internet will help support the concept of smart sensors, smart structures, and agent-based systems to be discussed in this report. Table 3-1 below compares the Web, X-Internet and Extended Internet along various dimensions. In an article related to the X-Internet in The Industry Standard, Thompson [7] cites Forrester as claiming that by 2010, there will be 14, 266 million Internet-based devices with 13,537 million being X-Internet and extended Internet devices.

Table 3-1: From Web, to X Internet to Extended Internet [8]

	The Web	X-Internet	Extended Internet
Number of devices	Millions	Hundreds of Millions	Billions
Focus	Browsers	User-focused software	Devices
Important Applications	Web, instant messaging, e-commerce	Responsive experiences	Real-time business applications
Data	HTML, XML	Executables and XML	Environmental data
Model	Server-centric	Peer-to-peer	Device-centric
Connections	User-driven	Opportunistic	Opportunistic
Time Frame	1993 to 2001	2001 onward	2005 onward

With this much intelligence and knowledge in the distributed heterogeneous network infrastructure, one can envision various types of devices, such as sensors, actuators, and controllers, organized into complex goal-driven systems that can be used for commerce, education, man-machine interactions, as well as NS/EP situations.

3.2 Gartner Group's Enterprise Nervous System Concept

Gartner has introduced the concept of the Enterprise Nervous System (ENS) [9], in which the ENS is an intelligent network that connects people, application systems and devices – possibly distributed at various locations, within different business units, and using diverse systems – in a real-time pro-active virtual enterprise. Although the ENS is an evolutionary step in system concepts, it is revolutionary in elevating the network to a new level, in that much more of the intelligence will reside in the network, as well as in the applications. "The ENS offloads logic from the application systems by supplying higher 'quality-of-service' communication, transforming messages, redirecting messages as appropriate (using logical business rules), and sometimes even tracking and controlling business processes." [9]

3.3 Internet-based Sensors

This section discusses the general concepts, architectures and standards for connecting sensors to the Internet. Transducers, which can be either sensors or actuators, are used in manufacturing, industrial control, automobiles, aerospace, civil engineering, biomedicine, etc. Transducer manufacturers are seeking to build low-cost, networked smart transducers.

In order to interface smart transducers with diverse networks with their associated protocols, the Technical Committee on Sensor Technology of the Institute of Electrical and Electronics Engineers (IEEE) Instrumentation and Measurement Society has sponsored a series of projects, called IEEE P1451, Smart Transducer Interface for Sensors and Actuators. In addition, NASA and the Jet Propulsion Lab are working on the Sensor Web, which is presented in Section 3.3.4. The Sensor Web consists of communities of low-power sensors that communicate wirelessly, and upload sensor data to control nodes connected to the Internet.

3.3.1 The IEEE P1451 Standards Family

"The goals of the IEEE P1451, Smart Transducer Interface for Sensors and Actuators, projects are to define a set of common communication interfaces for connecting transducers to microprocessor-based systems, instruments, and field networks in a network independent environment [10]."

The P1451 project family includes:

- P1451.1 – Network Capable Application Processor (NCAP) Information Model. This project is developing the NCAP object model for the components of a networked smart transducer, together with the software interface specifications to the components. The networked smart transducer object model provides interfaces for the NCAP transducer block which encapsulates the details of the transducer hardware implementation within a simple programming model, thereby making the sensor or actuator hardware interface appear to be an input-output driver. The interface to the NCAP block hides the details of different networking protocols within a small set of communications functions. The goal of having a common network-independent application model is that it enables a "plug-and-play" capability between the sensors/actuators and the target networks. P1451.1 [11] became an IEEE Standard in 1999.
- 1451.2 – Smart Transducer Interface Module. This project defines the Transducer Electronic Data Sheet (TEDS) together with its data format, the digital interface and communication protocols between the transducers and the microprocessor. The TEDS is stored in non-volatile memory associated with the transducer and contains information describing the transducer's type, attributes, operation, and calibration. This TEDS feature allows for transducer self-identification to the system or network. Thus a TEDS represents the meta-data that describes the transducer to the system and network. Each such transducer has its own TEDS. 1451.2 became on IEEE Standard on September 26, 1997 [12].
- 1451.3 – Transducer Bus Interface Module. This project aims to develop digital communications and TEDS formats for distributed multi-drop systems. The idea is to develop a protocol by which several physically separated transducers can be configured in a multi-drop configuration and automatically negotiate with the network.
- 1451.4 – Mixed-Mode Transducer. This proposed standard [13] has as its goal to connect analog transducers such as those used to monitor the condition of machinery by means of analog transducers such as piezoelectric transducers, piezoresistive transducers and strain gauges. Mixed-mode means transmitting transducer self-identification data in digital mode and then switching over to analog mode for outputting analog signal. Manufacturers would like to connect

such transducers to the network using two wires for analog signal transmission to communicate the digital TEDS data to instruments or NCAPs.

The goal of the standard is to:

- Allow "plug-and-play" capability at the transducer level by providing a common IEEE 1451.4 transducer communication interface that is compatible with legacy transducers,
- Simplify the creation of smart transducers,
- Simplify the setup and maintenance of instrumentation systems,
- Enable the bridging of legacy instrumentation with the smart mixed-mode transducers
- Enable the implementation of smart transducers with minimal use of memory.

Figure 3-1 shows the components of the mixed-mode transducer and interface t the network.

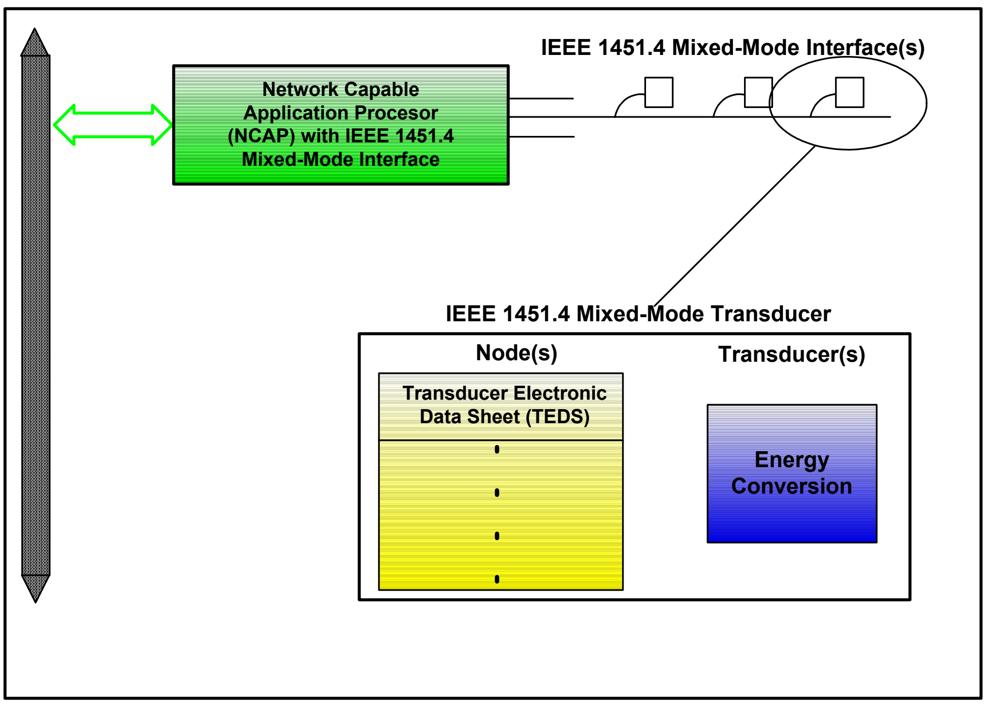

Figure 3-1: IEEE 1451.4 Context for Mixed-Mode Transducer and Interface [13]

3.3.2 P1451 Activities at NIST

The National Institute for Standards and Technology (NIST) has implemented a demonstration that uses the Java programming language and the Internet to monitor and control remotely a collection of IEEE 1451.2 networked smart transducers in a manufacturing environment. This is a demonstration of the high-speed Makino Machining Center at NIST which requires consistent coolant temperatures during the machining processes to reduce the amount of tool wear and to produce more exacting parts. An application of the IEEE 1451.2 Smart Transducer Interface Module (STIM) and the IEEE P1451.1 Network Capable Application Processor (NCAP) to the high-speed machining process has been developed and deployed across the Internet. This demonstration consists of a Java Applet that is capable of monitoring and controlling the coolant temperature control loop by an operator[2].

3.3.3 Bluetooth Technology for Wireless Transducer Networks

Dunbar [14] discusses the role of the wireless transducers communicating using the open Bluetooth protocol. He presents the CrossNet architecture that enable users to create wireless sensor networks. CrossNet allows hundreds of sensors of different brands and types to connect with data-acquisition systems such as handheld devices, and Internet-based personal- and desktop computers. Figure 3-2 shows how sensors and transducers can communicate using Bluetooth protocol to a local area network (LAN) in the Crossnet architecture.

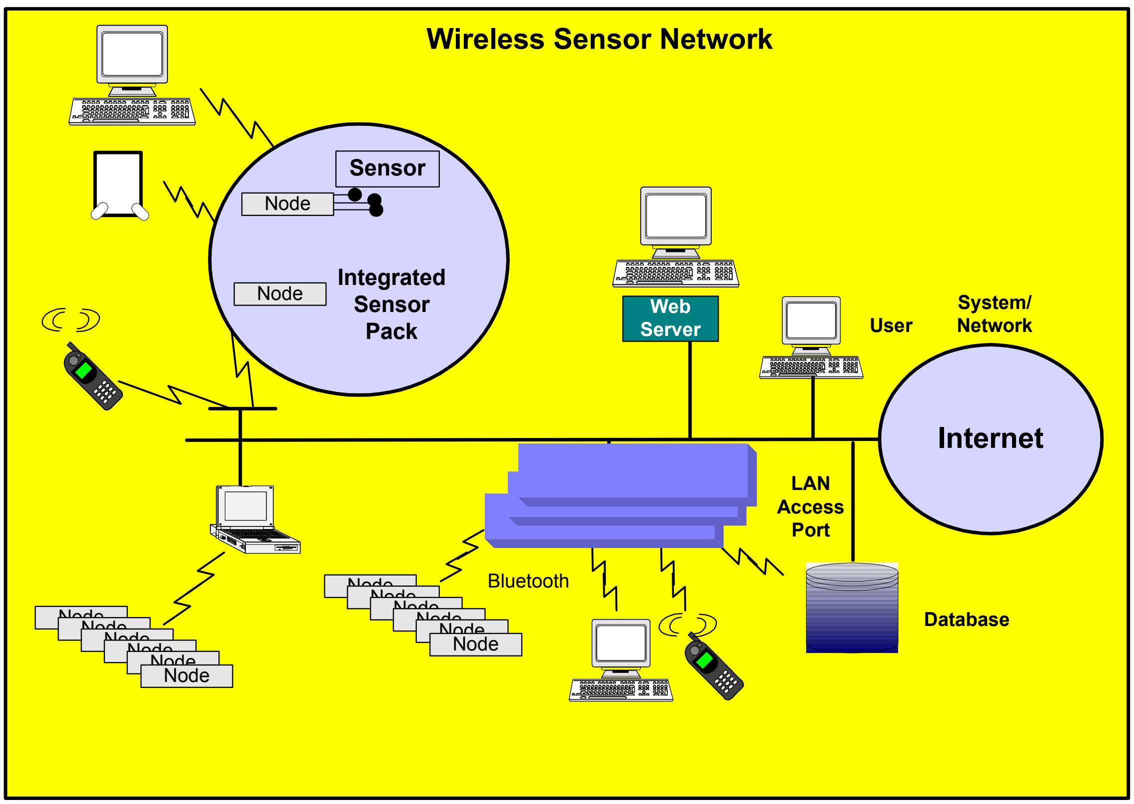

Figure 3-2: Architecture for Wireless Sensor Network based on CrossNet [14]

3.3.4 NASA-JPL Sensor Web

The National Aeronautics and Space Administration (NASA), in conjunction with Cal Tech's Jet Propulsion Lab (JPL) is developing a web of sensors, called Sensor Web, that eventually will be used to explore for signs of life on Mars, and other heavenly bodies

---

[2] See http://www.motion.aptd.nist.gov/P1451/ISADemo.htm

such as Jupiter's moon, Europa[3]. The Sensor Web consists of a collection of sensors contained in pods, which can be deployed over large areas. The pods communicate among themselves using wireless technology, and share information by hopping the data, either sequentially or in parallel, until the data reaches a prime node which serves as an uplink point to a computer connected to the Internet. Two types of pods have been developed, the gumball-sized MicroPod (see Figure 3-3) and the sandwich-sized Botanical Pod [15]. The Botanical pod has six on-board sensors: light level, oxygen level, humidity, soil moisture, and two for temperature. It is power by solar energy, and data points are taken every five minutes. The MicroPod weighs 50 grams, and has two on-board sensors to monitor temperature and light level. It takes measurements every second, consumes 50 microwatts of energy, and is powered by a 3-volt Lithium battery contained in its base.

Figure 3-3: Gumball-Sized Micro Pod [15]

On May 18, 2000, in the first test of the wireless-networked sensor-pod technology, NASA and JPL placed twelve Botanical Pods over sections of the Huntington Botanical Gardens. "The sensors collected synchronized data on moisture, temperature, sunlight, humidity and oxygen levels every five minutes. The solar-powered stations then regularly shuttled the data among themselves, eventually "hopping" it back to a primary pod connected to a laptop computer" [16]. The experiment lasted for 22 weeks.

[3] A Quicktime movie depicting the concept of deploying the Sensor Web can be found at: http://sensorwebs.jpl.nasa.gov/movies/SensorWebslg.mov

In addition to using the Sensor Web for space exploration, JPL envisions applications for agriculture, tracking and monitoring seismic activity, tracking the flow of toxins in ground water, underwater tracking for global warming experiments, in-situ measurements for weather prediction, and monitoring of civil structures where temperature and humidity are a concern. The key is that the Sensor Web uses smart sensors that consume very little power, are able to communicate among themselves intelligently using advance protocols, and connect to Internet-based controllers that can process the information.

4 Smart Structures

This section presents an overview of the concept of smart structures and materials and discusses several R&D efforts into smart structures taking place at universities and research labs. The focus is on those technologies and projects that could support NS/EP requirements and situations. The term smart structures can refer to smart materials, intelligent materials, active materials, adaptive materials as well as actuators and sensors [17].

4.1 Overview of Smart Structures

The field of smart materials and structures has shown technological innovations in engineering materials, sensors, actuators, and image processing [18]. Smart materials are defined in many different ways:

- Materials that function as both sensors and actuators,
- Materials which have multiple responses to one stimulus and act in a coordinated fashion,
- Passively smart materials with self-repairing or stand-by characteristics to withstand sudden changes,
- Actively smart materials utilizing feedback,
- Smart materials and systems simulating biological functions in load bearing structural systems.

Jain [18] provides an analogy between smart structural systems and biological systems:

> *"The goal of smart structures technology is to reproduce biological functions in load bearing structural systems. These biological functions would include a skeletal system to provide load bearing capability, a nervous system which is a network of embedded or attached sensors to monitor the state of the structure, a motor system to provide adaptive response, an immune system to provide healing capability, and a neural system to provide learning and decision making functions."*

Some necessary characteristics for actuators and sensors are [18]:

> *"Sensor materials should have the ability to feedback stimuli such as thermal, electrical and magnetic signals, to the motor system in response to changes in the thermomechanical characteristics of the smart structures. Whereas, actuator materials should have the ability to change the shape, stiffness, position, natural frequency, damping and/or other mechanical characteristics of the smart structures in response to changes in temperature, electric field and/or magnetic field. The most popular material systems being used for sensors and actuators are piezoelectric materials, magnetostrictive materials, shape memory alloys, electrorheological fluids and optical fibers. Magnetostrictive materials, shape memory alloys and electrorheological fluids are used as actuator*

materials. Optical fibers, however, are used primarily as sensor materials. Among all these active materials, piezoelectric materials are most widely used because of their fast electromechanical response, low power requirements and relatively high generative forces."

The remainder of this section provides a summary of some types of smart materials and their properties.

4.1.1 Piezoelectrics

Piezoelectric materials [17] produce an electrical field when subject to a mechanical strain and conversely, if an electrical field is applied to them a deformation results. When strained they produce an electric field and act like a sensor for stress or strain sensing. When an electric field is applied, piezoelectrics behave as actuators. Perfect collocation can happen when the same piece of piezoelectric material is used for both sensing and actuating. In practice, however, and to avoid non-linearities due to interactions between actuating and sensing signals, two separate pieces of piezoelectrics are used for collocation. Piezoelectric actuators have been used for active shaping, as well as for vibration and acoustic control of structures. Their ability to be easily integrated into structures makes them very attractive in structural control because all the moving parts encountered with conventional actuators are eliminated. These materials could be used in NS/EP situations where critical infrastructures such as bridges could be stabilized using them to detect unusual movements and sending signals to warning or control systems.

4.1.2 Shape Memory Alloys

Shape memory alloys (SMA) [17] are a type of adaptive material that convert thermal energy directly into mechanical work. A variety of alloys, when properly mixed, exhibit this effect by repeated heat treatments. One-way shape memory effect occurs when one of these special alloys is mechanically deformed at low temperatures, and then heated above a critical transition temperature thereby restoring the specimen's original "memory" shape. Some SMAs can be trained to exhibit a two-way shape-memory effect whereby heating the SMA results in one memorized shape while cooling results in a second different shape [19]. Shape memory alloys are highly adaptive, compact, light weight, and have a high force-to-weight ratio [19]. For example, a research project at the Technical University of Berlin[4] uses SMAs to shape an airfoil. Shape memory wires contract when they are heated and they expand when they are cooled by the surrounding air flow. An electric current will make the wire contract, because of Joule heating, and cooling by the surrounding air will make it expand, when the current is switched off. This phenomenon can be used to adapt the shape of an airfoil to the prevailing flight conditions. A Java-based SMA simulator is available[5].

Rod MacGregor has invented the nanomuscle[6] made from a nickel-titanium alloy. SMAs made of nickel-titanium material are extremely elastic [20]. It is used in cell-phone

[4] See http://www.thermodynamik.tu-berlin.de/english/SMA/wing/wing.html

[5] See http://www.thermodynamik.tu-berlin.de/haupt/simulation/Sma_Sim_Home.html

[6] See http://www.nanomuscle.com/

antennas, and eyeglass frames, because it has the ability to remember its shape. On a molecular level, the alloy has a cubic lattice structure that can be stretched or twisted into a parallelogram, and then it retains that configuration. But when electricity is passed through it, the lattice structure snaps back into a cube with tremendous force.

MacGregor harnesses the power of SMAs by attaching nickel-titanium filaments to small plates about 5 centimeters long that are stacked on top of one another and move back and forth. To make the nanomuscle move consistently and reliably, MacGregor uses an embedded 8-bit microprocessor the size of a sunflower seed and sensors to ensure the alloy is moving exactly as planned [20]. Hasbro, the toy maker, will introduce nanomuscles into its toy line in 2002, and has options for 10 million nanomuscles.

4.1.3 Magnetostrictive

Magnetostriction is a property of certain metallic materials where there is an increase or decrease in strain when a specimen is subjected to magnetic fields. Conversely, the material generates a magnetic field when stressed. This phenomenon was first discovered in 1842 by J.P. Joule in nickel. Subsequently, cobalt, iron and alloys of these materials were found to show a significant magnetostrictive effect. The most promising magnetostrictive material is Terfenol-D, an alloy of iron and the rare earth elements, Terbium and Dysprosium. Terfenol-D is well suited for a number of sensor and actuator applications including: sonar transducers, anti-vibration systems, sonic sound-sources, micro-positioning, and stress sensors.

4.1.4 Electrorheological Fluids

Electrorheological Fluids (ER) materials are suspensions that undergo reversible changes in rheological properties such as viscosity, plasticity and elasticity when subjected to electric fields. These reversible changes are due to the controllable interaction between micron-sized dielectric particles within the ER suspension. The polarization of these particles leads to configuration changes, which in turn results in significant changes in rheological properties. When an electric field is applied, the ER material is transformed from that of a liquid to that of a solid-like gel.

ER material applications can be categorized in two classes, controllable devices and adaptive structures. ER based controllable devices include, valves, mounts, clutches, brakes and dampers. Adaptive structures are those with tunable properties due to the incorporation of a controllable component such as an ER material. ER fluids have been studied for active control of flexible rotor blades.

4.2 Smart Interfaces, Buildings, and Machines

This section reviews research on haptic interfaces and discusses how they are relevant to intelligent systems analysis, design, and simulation, as well as their role in support of NS/EP requirements. It also focuses on the use of smart structures and materials in the context of feedback control systems for buildings and the use of micro-machines. The goal for smart structures and materials research is to incorporate these materials into existing materials, or to augment existing structures with smart materials so as to control their behavior.

4.2.1 *Haptic Interfaces*

"Haptic interfaces are robots that provide information to people by manipulating them", according to Hollerbach [21]. They are devices that allow users to interact with virtual environments or teleoperated systems. A haptic interface allows one to "feel" the objects in virtual world, or a faithful representation of that object at a distant location.

Haptic exploration is a sensory task, which identifies surface or volumetric properties. The human tactile sensory system consists of tactile and kinesthetic information. For tactile information, mechanoreceptors in the finger pad play the major role, while for the kinesthetic information, sensory receptors in the skin around the joints, joint capsules, tendons, and muscles play the major role.

Both Academe and Industry have explored interfaces that combine both sensory information sets. The body-based and ground-based Haptic interfaces excite human kinesthetic sensors, while tactile displays excite tactile sensors. Dr. John Hollarbach of the University of Utah has identified several research areas [21] for haptic interfaces. They are:

- Workspace and dexterity. This involves choosing the correct haptic interface for the task at hand.
- Building accurate, interactive simulations. Here the challenge is to model the forces arising from the interaction of the user with the haptic object in a simulation. Also important is modeling the interaction among objects in a simulated environment.
- Scientific visualization. In the real world, we have vast quantities of data that can best be understood by "visualizing" the data, that is, by presenting it in ways that can be understood by humans. Haptic interfaces such as 3D goggles can allow a person to enter the data space and see various phenomena. Examples include viewing blood flow through the heart. Hollerbach's team at the University of Utah has used visualization to "probe heterogeneous finite element mesh-fitting errors, where the amount of error at a particular mesh element was played back as a vibration".
- Locomotion interfaces. The interfaces overcome the limitations of using joysticks for maneuvering. The main types of locomotion interfaces include bicycles and unicycles, treadmills programmable foot platforms, and walk-in-place systems.

Haptic feedback can be used in virtual reality environments where more information can be conveyed by touch than by just sight and sound, for example haptic interfaces can be used to control robots and the feel of the object in a robots hand can provide the operator with feedback so as to prevent crushing the object. This technology can be used for NS/EP hazardous situations where robots, under the direction of human operators using haptic interfaces, can control the robots to obtain a feel for the severity of structural damage, etc. Haptic feedback is also being used to train surgeons and to perform remote surgery. A virtual scalpel can be controlled, via a haptic interface, allowing the user to operate on a virtual patient. In the case of real surgical procedures, a tele-surgeon, in conjunction with a surgical "robot" could operate on patients. Here again, this technology could be used in NS/EP situations where expert medical staff could operate at a distance. Haptic feedback could also be used in keyhole surgery [22]. Here, tiny cameras and surgical instraments are inserted via tiny incisions in the body of the patient. Fiber optic cables are used to link the cameras with TV screens so that the surgeon can see inside the patient, for example to perform scull surgery using an endoscope [23].

Researchers at MIT and SensAble Technologies [24] have developed the Phantom haptic interface that lets one feel virtual objects with one's hands. They have also developed the General Haptics Open Software Toolkit (GHOST) that allows users to create objects and interact with them. One application of the Phantom/GHOST system is the virtual workbench for training electronic technicians, jointly developed by MIT and the Naval Air Systems Division (NAWC/TSD). This Virtual Environment (VE) testbed "uses a semisilvered mirror so that the virtual visual image of an electronic circuit board lies within the Phantom interface's workspace. Trainees can see the circuit board along with their own hand on it, feel the components of the circuit board with a probe, use a virtual multimeter at various contact points and even haptically operate switches on it to observe changes in the circuit's electrical behavior." [24]

4.2.2 Smart Buildings and Bridges

Spencer and Sain of the University of Notre Dame [25] have noted that the "protection of civil structures, including their material contents and human occupants, is without doubt a world-wide priority of the most serious current importance. Such protection may range from reliable operation and comfort, on the one hand, to survivability on the other. Examples of such structures include buildings, offshore rigs, towers, roads, bridges, and pipelines. In like manner, events which cause the need for such protective measures are earthquakes, winds, waves, traffic, lightning, and today, regrettably, deliberate acts. Indications are that control methods will be able to make a genuine contribution to this problem area, which is of great economic and social importance."

Spencer and Sain [25] also discuss the use of an active mass damper (AMD) system to control the motion of large buildings. In order to control the structure a small mass, usually less than 1% of the total mass of the structure, is installed on one of the upper floors of the structure, and an actuator is connected between the auxiliary mass and the structure. Loads and responses are measured at key locations in the building and are fed to a computer control system depicted in Figure 4-1.

The control system works as follows. The computer processes the responses according to a control algorithm and sends an appropriate signal to the AMD actuator, which in turn reacts against the auxiliary mass, applying inertial control forces to the structure so as to reduce the structural response in the desired fashion. Various types of control strategies have been used in the active control of civil engineering structures [25], including sliding mode control, fuzzy control, neural control, and reliability-based control, to name a few. The authors [25] cite that in 1989, the Kajima Corporation, was the first to use active control on the Kyobaski Seiwa building in Tokyo, Japan. Both active and hybrid control strategies have been used in building and bridges, primarily in Japan which is prone to earthquakes. Projects include the Kansai International Airport's control tower, and the Hiroshima Reihga Royal Hotel, which is 150 meters tall and has 35 stories.

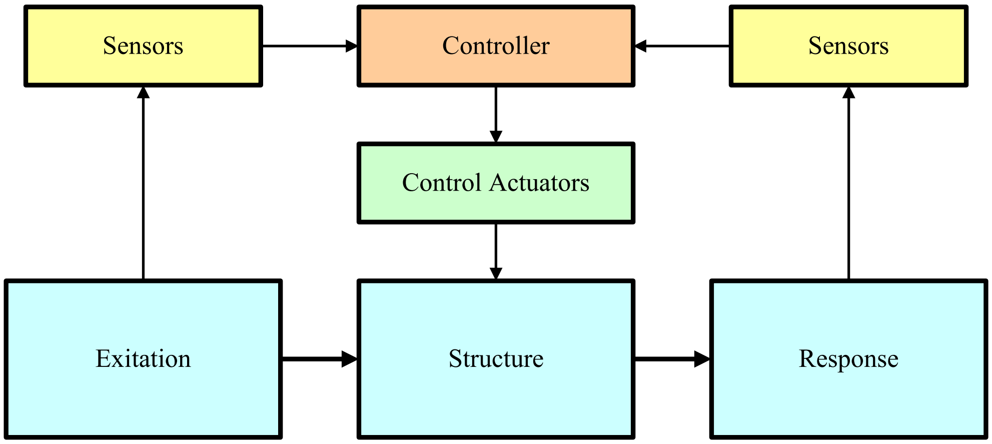

Figure 4-1: Schematic of the Structural Control Problem [25]

In civil engineering structures subjected to strong earthquakes and severe winds, passive and active control systems are at opposite ends of the spectrum for using supplemental damping for response reduction. Researchers at Notre Dame are using semi-active control systems, which combine the best features of both approaches, by offering the reliability of passive devices, while maintaining the versatility of fully active control systems. They use magnetorheological (MR) dampers that use MR fluids to create controllable dampers [26]. They state, "Initial results indicate that these devices are quite promising for civil engineering applications. They are capable of generating large forces, they offer highly reliable operation at a modest cost and can be viewed as fail-safe in that they become passive dampers should the control hardware malfunction."

4.2.3 Micromachines

Micro Electro Mechanical Systems (MEMS) are micro-machines that have both mechanical and electronic structures built in silicon. They are fabricated using integrated circuit (IC) batch processing techniques and can range in size from micrometers to millimeters. These systems can sense, control and actuate on the micro scale, and function individually or in arrays to generate effects on the macro scale.

One important property of MEMS devices is their ability to host multiple transducers in the same device. This method can be used to increase the dynamic range of the device without having to engineer a single transducer to cover the entire dynamic range with the same sensitivity and resolution.

Many potential applications have been proposed for MEMS including:

- accelerometers, pressure, chemical and flow sensors;
- micro-optics; optical scanners;
- fluid pumps;
- inertial measurements;
- environmental measurements for weather forecasting;

- chemical detection for pollution analysis and compliance;
- biotoxin detection;
- seismometry for space science and oil exploration;
- blood pressure monitoring;
- in-vitro drug delivery;
- telecommunication switches;
- fluid flow controllers; and
- active flight control surfaces.

Sandia National Labs has a web site devoted to intelligent micromachines[7], and W. David Williams, Director, Microelectronics and Photonics, Sandia National Laboratories, states:

> *"To achieve the vision of an intelligent, integrated microsystem, we must capitalize on key developments which have been made, and others which still need to be made, in microsensors, logic circuits, micromachines, and advanced communications (wireless - optical and /or microwave). The applications of these devices, from medical treatment, to anti-terrorism, to the health monitoring of complex engineering systems may revolutionize our future world."*

Sandia's state-of-the-art fabrication technologies are available to everyone through the SAMPLES (Sandia Agile MEMS Prototyping, Layout Tools, Education, and Services) Program. The Sandia Web Site has additional links describing its research program in MEMS [27]. The site also contains images and videos showing MEMS in action[8].

There are several notable MEMS research centers. They are:

- The Berkeley Sensor and Actuator Center (SAC)[9]
- The DARPA MEMS Research Program[10]
- The IDA MEMS Program[11]
- The SANDIA Liga Micromachine Program[12]
- The Quate Group at Stanford University[13]
- The University of Wisconsin – Madison[14]
- The Semiconductor Subway[15]

---

[7] See http://www.mdl.sandia.gov/1300vision.html
[8] See http://www.sandia.gov/mems/micromachine/movies.html
[9] See http://bsac.eecs.berkeley.edu/
[10] See http://www.darpa.mil/MTO/MEMS/
[11] See http://www.ida.org/MEMS/
[12] See http://daytona.ca.sandia.gov/LIGA/
[13] See http://www.stanford.edu/group/quate_group/index.html
[14] See http://mems.engr.wisc.edu/.
[15] See http://www-mtl.mit.edu/semisubway.html

5 Intelligent Agent-Based Systems

This section provides an overview of intelligent agent concepts. It also provides definitions of agents, and compares objects, agents and components. It provides a taxonomy of the different kinds of agents, including mobile, autonomous, cooperating, and negotiating agents, and describes some agent models.

Intelligent systems originally emerged from the artificial intelligence community. There was a great deal of interest in knowledge-based systems, which are rule-based intelligent systems (also referred to as expert systems). These systems were considered flexible because rule bases could be easily modified. However, in practice these early systems were often monolithic rule-based systems, and as the intertwined rule base grew in size with often over 1000 inter-dependent rules, it often became difficult to understand and maintain. More recently, a much more flexible, distributed and adaptable paradigm has emerged for intelligent systems, namely that of intelligent agent-based systems.

5.1 Agents

There are many definitions of agents [28]. The dictionary definition of agent is "one that is authorized to act for or in the place of another as a representative, emissary, or a government official". From the standards organizations, Object Management Group (OMG) defines agents as "computer programs acting autonomously on behalf of a person or organization"[16]. Foundation for Intelligent Physical Agents (FIPA), another standards organization, defines agents as "computational processes implementing an application's autonomous communicating functionality"[17].

Agents vary considerably in their capabilities. While much of the early research was in artificial intelligence, the growth of the Internet has emphasized distributed computing. Java, which has many features for distributed computing [29] has become a popular language for programming agent-based systems. Distributed Java technologies, in particular Remote Method Invocation (RMI) and Java Intelligent Network Infrastructure (Jini), have provided the distributed application infrastructure for distributed agent-based systems implemented in Java.

An intelligent agent-based system is a highly distributed system, in which agents are active concurrent objects that acting on behalf of users. Agents are intermediaries between clients, which are typically user interface objects, and servers, which are typically entity objects that store persistent information. Usually, several agents participate in the problem solving activity, communicating with each other. This leads to a more distributed and scaleable environment. More information on multi-agent systems is given in [30-39]. Several researchers have addressed interaction protocols in multi-agent systems [40-45].

---

[16] See: http://www.omg.org/

[17] See: www.fipa.org

5.2 Categorization of Agents

Agents can be categorized in different ways, based on their mobility, based on their intelligence, or based on the roles they play in an agent-based system. One categorization is whether the agent is stationary or mobile. Another categorization is based on the following capabilities [33, 45]:

- Cooperative agents communicate with other agents and their actions depend on the results of the communication,
- Proactive agents initiate actions without user prompting, and
- Adaptive agents learn from past experience.

Agents may combine the above three capabilities [46], as shown in Figure 5-1. Personal agents are proactive and serve individual users – they may also be adaptive. Adaptive personal agents can search for user information in background mode and are often coupled with the World Wide Web. They can refine their search strategies based on how the user reacts to previous searches. Collaborative agents are both proactive and cooperative.

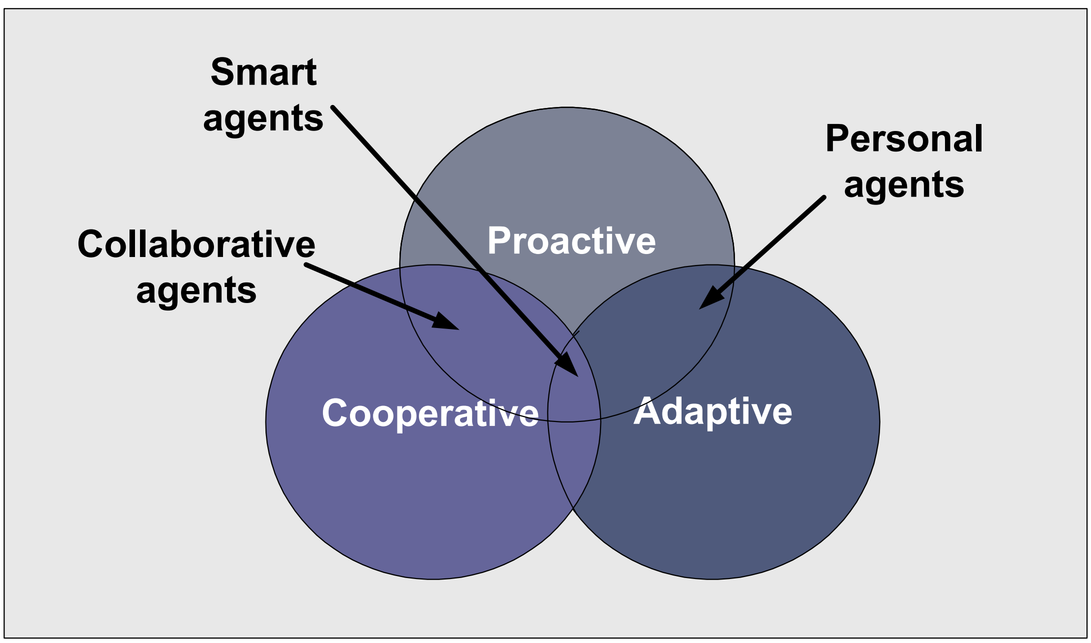

Figure 5-1: Categorization of Agents [46]

Knowledge Rover Architecture

Another categorization of agents was carried out as part of the Knowledge Rover Architecture, developed in support of the Defense Advanced Research Projects Agency (DARPA) Advanced Logistics Program [33, 34, 47]. Knowledge rovers are cooperating intelligent agents that can be configured automatically with appropriate knowledge bases (ontologies), task-specific information, negotiation and communication protocols for its mission into cyberspace. The different agents are:

Coordinator Agent — coordinates the activities of a group of agents. It is informed of significant events. A significant event can lead to the activation of new agents. For example, if the enterprise is notified of disaster-relief request, then the coordinator agent would coordinate with other agents in implementing the relief scenario.

User Agents — acts on behalf of a user, and is responsible for assisting users: 1) in browsing catalogs and information holdings such as the information repository, 2) in the intelligent formulation of queries, and 3) in the planning of tasks within a mission-specific scenario such as provisioning logistic support for a disaster relief effort.

Real-time Agents — are mission-specific, defined and configured to process incoming data, and update the appropriate database or notify the appropriate users. The real-time agents are autonomous, communicate with each other using a pre-defined protocol. Real-time agents are responsible for monitoring the external environment, interacting with other systems, or acting on inputs from users. When an event is detected by a real-time agent, it is signaled to the relevant agents.

Facilitation Agents — provide intelligent dictionary and object location services. For example a facilitation agent [31] might accept a request from the Coordinator Agent to find all *external* providers of 'antibiotics,' and it might respond with the pharmaceutical producers and suppliers for the region in question. Other agents such as knowledge rovers (defined below) could then arrange for the items to be requisitioned, retrieved, and paid for. A knowledge rover could also post a request for bids, accept responses, make contracts, and provision the requested items.

Mediation Agents — are configured to assist in the integration of information from multiple data and information sources, having diverse data formats, different meanings, differing time units, and providing differing levels of information quality. Mediators [48, 49] are configured to accept queries from the Coordinator Agent, translate the queries into the query language of the appropriate database system, accept the retrieved result, integrate it with results from other sources, and return the information to the Coordinator Agent for presentation to the User Agent.

Active View Agents — are created to monitor real-time events from the environment or from databases, and to use these events to initiate actions that will result in the update and synchronization of objects in the Data Warehouse and also in local views maintained at user workstations. These agents are configured to perform very specialized monitoring tasks and have a collection of rules and actions that can be executed in response to events and conditions that occur in the environment or in the databases.

Information Curators — are responsible for the quality of information the Information Repository. They assist in evolving the data and knowledge bases associated with enterprise information resources. They work with knowledge rovers to incorporate newly discovered resources into the information repositories.

Knowledge Rovers — are instructed to carry out specific tasks on behalf of the coordinator agent, such as to identify which vendors have a specific item on hand. This would involve obtaining information from several vendors. The knowledge rover dispatches field agents to specific sites to get the relevant information. If the knowledge rover gets similar information from more than one source, it may ask a mediator resolve the inconsistency. The knowledge rover reports back to the Coordinator Agent. The rovers are also responsible for Internet resource discovery. These new information sources and their data are analyzed to determine the adequacy, quality and reliability of retrieved information and whether it should be incorporated into the information repository.

Field Agents — are specialized rovers that have expertise in a certain domain, for example, pharmaceuticals, and knowledge about domain-specific information holdings at one or more sites. For example, a field agent could be tasked to monitor all aspects of a single item, say an 'antibiotic' produced by several manufactures and distributed by several vendors. They negotiate with the local systems, retrieve appropriate data, and forward it to the appropriate requesting agent.

5.3 Agent Communication Languages

An important issue in the development of multi-agent systems is interoperability. There is a need for agents built by different organizations using different hardware and software platforms to communicate with each other via a common language with a universally agreed-upon semantics. This requirement has led to the development of agent communication languages (ACL) [50].

One of the most widely referenced ACLs is the DARPA-sponsored Knowledge Query and Manipulation Language (KQML) and the associated Knowledge Interchange Format (KIF), which is a common framework through which multiple intelligent systems can exchange knowledge [51-53]. KQML is an "outer" language for messages. It defines a format for messages and supports 41 performatives, which are message types that define what is intended by the messages. KQML does not address the content of the messages. This is done by KIF, which is a classic first-order predicate logic language converted to a LISP-like syntax.

Another language is the FIPA work on an agent communication language called ACL [54]. ACL also defines an "outer" language for messages with 20 performatives. The FIPA ACL has been given a formal semantic foundation. The two languages are quite similar [50].

An example of a KQML statement in an NS/EP setting concerning toxic chemicals is given below. The *performative* is **ask-one** in which an agent asks one question of another agent and expects exactly one reply. The *content* of the message is a query about the toxic chemical level of cyanide. The *receiver* is the toxic chemicals agent who receives the message. The receiver is assumed to understand the *language* in which the content is expressed, namely LPROLOG. The toxic chemicals *ontology* defines the terminology used in the message.

```
(ask-one
    : content (TOXIC-CHEMICAL-LEVEL CYANIDE ?toxicity-level)
    : receiver toxic-chemicals-agent
    : language LPROLOG
    : ontology toxic-chemicals
)
```

In the above example, after a train wreck emergency resulting in the release of toxic cyanide, the user agent sends the toxic chemical agent a request for the toxic chemical level of cyanide. The agent determines the toxicity level and returns the value to the client agent.

Compare the above with a FIPA statement for the same query, which has quite similar syntax.

```
(inform
        : sender my-agent
        : receiver toxic-chemicals-agent
        : content (TOXIC-CHEMICAL-LEVEL CYANIDE ?toxicity-level)
        : language LPROLOG
        : ontology toxic-chemicals
)
```

The ACLs allow a great deal of flexibility in communication between agents. The ACLs allow different agents to communicate using different languages and ontologies. By providing a choice of language and ontology, they allow agents to communicate using a protocol best suited to their needs. The ACLs allow for future evolution to accommodate new languages and ontologies.

5.4 Inter-Agent Communication

5.4.1 Message Communication in Cooperative Agent-Based Systems

Agents may communicate with each other in many different ways, either directly or indirectly. Client agents act on behalf of their clients while server agents provide agent services for the client agents. Server agents may register their services with an object broker. Client agents locate server agents via the object broker. A Client Agent may also subscribe with a server agent to receive events of a given type. When an event of that type occurs, the Server Agent multicasts the event to all clients on the subscription list.

A designer of a distributed intelligent agent-based system, where distributed objects potentially reside on different nodes, needs to consider a wide variety of alternatives for message communication [55, 56]:

- *Synchronous message communication.* Also referred to as a Remote Procedure call, can be either in the form of single client/server communication or multiple client/server communication. In both cases a client sends a message to the server and waits for a response; in the latter case a queue may build up at the server. In a client/server architecture, it is also possible for a server to delegate the processing of a client's request to another server, which then responds directly to the original client.
- *Asynchronous message communication.* Asynchronous message communication is accomplished either by means of first-in-first-out (FIFO) message queues or priority message queues. In distributed environments, asynchronous message communication is used wherever possible for greater flexibility. This approach can be used if the sender does not need a response from the receiver.

Alternatively, if it does not need an immediate response, the sender can receive it later.

- *Connections.* If the client and server are to have a dialog that involves several messages and responses, a connection may be established between the client and the server. Messages are then sent and received over the connection.
- *Subscription/notification communication*. With this form of group communication, one message is sent to several recipients, e.g., from the server to its clients. Group communication may be broadcast or multicast communication. With broadcast communication, an unsolicited message may be broadcast to several clients, e.g., informing them of a pending shutdown. The client then decides whether to act on the event or not. With multicast communication, clients may subscribe to events to which they wish to be notified. When an event of interest arises, a message is sent to all clients on the subscription list notifying them of the event. With group communication, one object acts as an event broker. This object receives the message to be multicast, reads the subscription list to determine which objects wish to be notified of this event, and then sends the message to all subscribers.
- *Transaction support*. For transactions that need to be atomic (i.e., indivisible), services are needed to begin the transaction, commit the transaction, or abort the transaction. This is typically used for updates to a distributed database that need to be atomic, e.g., transferring funds from an account at one bank to an account at a different bank. Using this approach, updates to the database are coordinated, such that they are either all performed (committed) or all rolled back (aborted).
- *Brokered communication*. An object request broker mediates interactions between clients and servers. It frees the client from having to maintain information about where a particular service is provided and how to obtain the service. Servers register their location and the services they provide with the broker. Clients can query the broker for available services.
- *Negotiated communication*. In a negotiated communication, agents will make proposals to each other, and respond with offers that may be accepted, rejected, or counter offered. This is discussed in more detail in Section 5.5.

For intelligent agent-based systems, the most flexible forms of communication are those that decouple agents from each other, in particular brokered communication and subscription / notification. With these approaches, agents can discover services provided by other agents after they have been deployed. They can register services they provide after deployment and can subscribe to services provided by other agents after deployment

5.4.2 Brokered Communication between Distributed Agents

In a distributed object environment, clients and servers are designed as distributed objects. An Object Broker [57] is an intermediary in interactions between client agents and server agents, and also in interactions between server agents and server entity objects. It frees client agents from having to maintain information about where a particular service is provided and how to obtain that service. It also provides *location* *transparency*, so that if the server agent is moved to a different location, only the object broker need be notified. A client agent sends a message identifying the service required, for example to request a videoconference service from a given service provider agent. The Object Broker receives the client request, determines the location of the service provider agent (e.g., the node the agent resides on), and forwards the message to the service provider agent at the specific location. This mode of communication, where the Client Agent knows the required service but not its location, is referred to as "white page" object brokering.

This type of brokering is referred to as a forwarding design and is depicted in Figure 5-2, which uses the Unified Modeling Language (UML) notation for documenting the object-oriented designs [55]. The service provider agent sends the response to the broker, which forwards it to the client agent [55, 56].

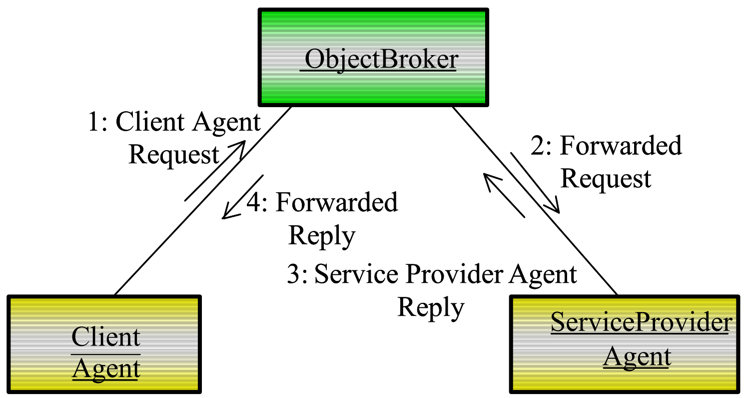

Figure 5-2: Object Broker Architecture (White pages - forwarding design)

Another type of brokering is the handle-driven design. In this case, the broker returns a service handle to the client agent, as shown in Figure 5-3 using the UML notation, which can then request the service at the server. The message arrives at the server agent, where the requested service is invoked.

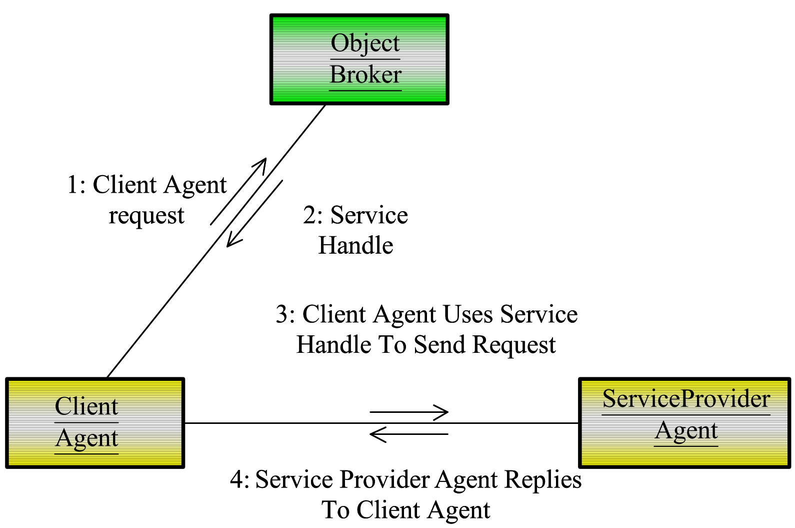

Figure 5-3: Inter-Agent Communication Using Object Broker (White Pages - handle-driven design)

The handle-driven design approach is as follows:

1. Client Agent sends a request to the Object Broker.

2. The Object Broker looks up the location of the server agent and returns a service handle to the client.

3. The Client Agent uses the service handle to request the service from the appropriate Server Agent.

4. The Server Agent services the request and sends the response directly to the Client Agent. This approach is more efficient if the Client and Server Agents are likely to have a dialog that results in the exchange of several messages.

Yet another type of brokering is through "yellow page" object brokering, where the Client Agent knows the service type it requires but not which specific Server Agent. This interaction is shown in Figure 5-4 using the UML notation. The Client Agent sends a

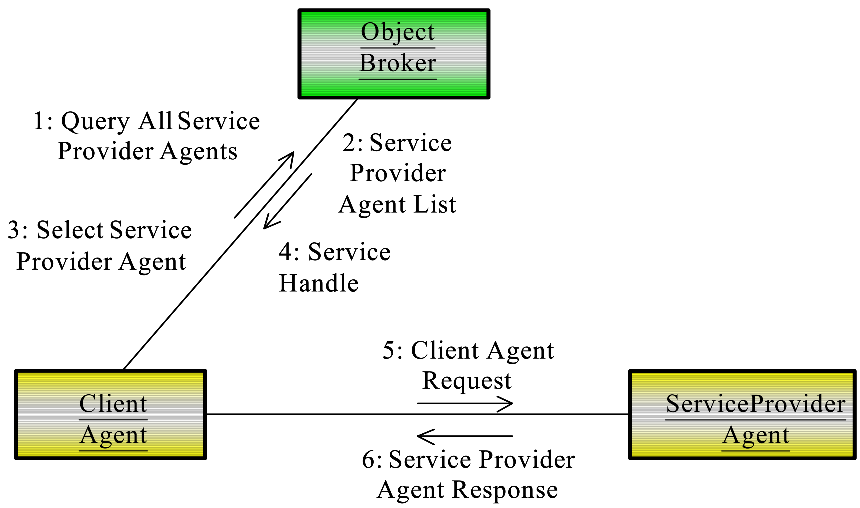

Figure 5-4: Inter-Agent Communication Using Object Broker (Yellow Pages)

query request to the Object Broker requesting all server agents of a given type, e.g., service provider agents. The Object Broker responds with a list of all service provider agents that match the client agent's request. The Client Agent, after consultation with the user, selects a specific service provider. The Client Agent then communicates with the service provider agent using the service handle returned to it by the broker. The interaction is described in more detail as follows:

1. Client Agent sends a "yellow pages" request to the Object Broker requesting information about all Service Provider Agents.

2. The Object Broker looks up this information and returns a list of all Service Provider Agents registered with it.

3. The Client Agent selects one of the Service Provider Agents and sends a "white pages" request to the Object Broker.

4. The Object Broker looks up the location of the Service Provider Agent and returns a service handle to the client.

5. The Client Agent uses the service handle to request the service from the appropriate Service Provider Agent.

6. The Service Provider Agent services the request and sends the response directly to the Client Agent.

5.4.3 Inter-agent Communication by Subscription and Notification

Information agents can cooperate using subscription and notification. An example is given of a Real-Time agent that monitors some external entity for specific external events. The Real-Time Agent maintains a subscription list of client agents who wish to be notified of these events. The agent may observe several events and must determine which of these is significant. When a significant event occurs to the external entity, the Real-Time agent updates an event archive and notifies client agents, via an Event Distributor object, who have subscribed to receive events of this type. The notification message is multicast to all the subscribers.

The Subscription and Notification inter-agent communication is depicted on the collaboration diagram using the UML notation (Figure 5-5), which shows three

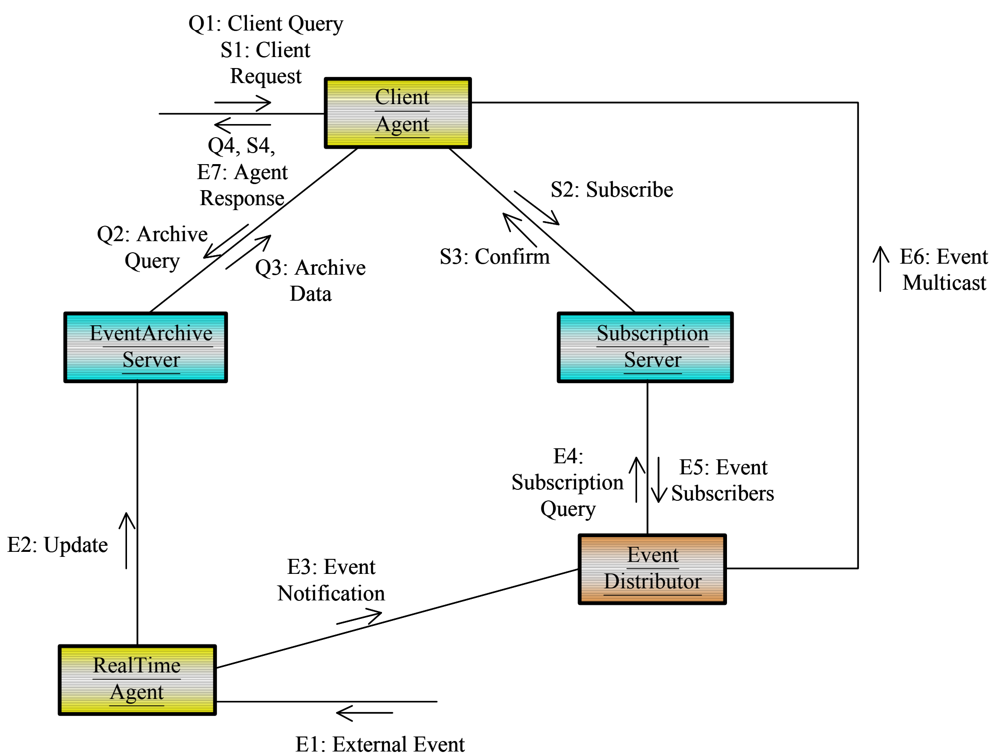

Figure 5-5: Inter-Agent Communication Using Subscription & Notification

collaborations, a simple query collaboration, an event subscription collaboration and an event notification collaboration. In the query collaboration (which does not involve a subscription) a client makes a request to the Client Agent, which queries the Event Archive Server and sends the response to the Client. The event subscription collaboration consists of the following interactions: The three event sequences are given different prefixes to differentiate between them:

- S: The event subscription collaboration consists of the following interactions (event sequence with S prefix)
- E: The event notification collaboration (event sequence with E prefix)
- Q: Query collaboration (event sequence with Q prefix)

The event subscription collaboration:

S1: Client Agent receives a subscription request from a user.

S2: The Client Agent sends a subscription message to the Subscription Server.

S3: The Subscription Server confirms the subscription.

S4: The Client Agent informs the user.

The event notification collaboration:

E1: An event arrives at the Real-Time Monitoring Agent.

E2: The Real-Time Monitoring Agent determines that this is a significant event and sends an update message to the Event Archive Server.

E3: The Real-Time Monitoring Agent sends an Event Notification message to the Event Distributor object.

E4, E5: The Event Distributor queries the Subscription Server to get the list of event subscribers (i.e., client agents that have subscribed to receive events of this type).

E6: The Event Distributor sends a multicast message to all client agents that have subscribed for this event.

E7: The Client Agent sends the relevant date to the client.

The query collaboration:

Q1: Client sends a query to the Client Agent, for example requesting events over the past 24 hours.

Q2: Client Agent forwards the query to the Event Archive Server.

Q3: Event Archive Server responds with the appropriate archive data, e.g., events over the past 24 hours.

Q4: Client Agent sends response to the client with the requested data.

Another example of an inter-agent collaboration strategy that is based on multicast communication is communication via the field [46]. With this approach, agents can subscribe and unsubscribe to the field independently of other agents. Each agent monitors the messages flowing in the field and reacts to them according to its criteria, as shown in Figure 5-6. The field is an event-driven multicast-based communication approach, similar to that used in the Subscription and Notification example described in the previous paragraph.

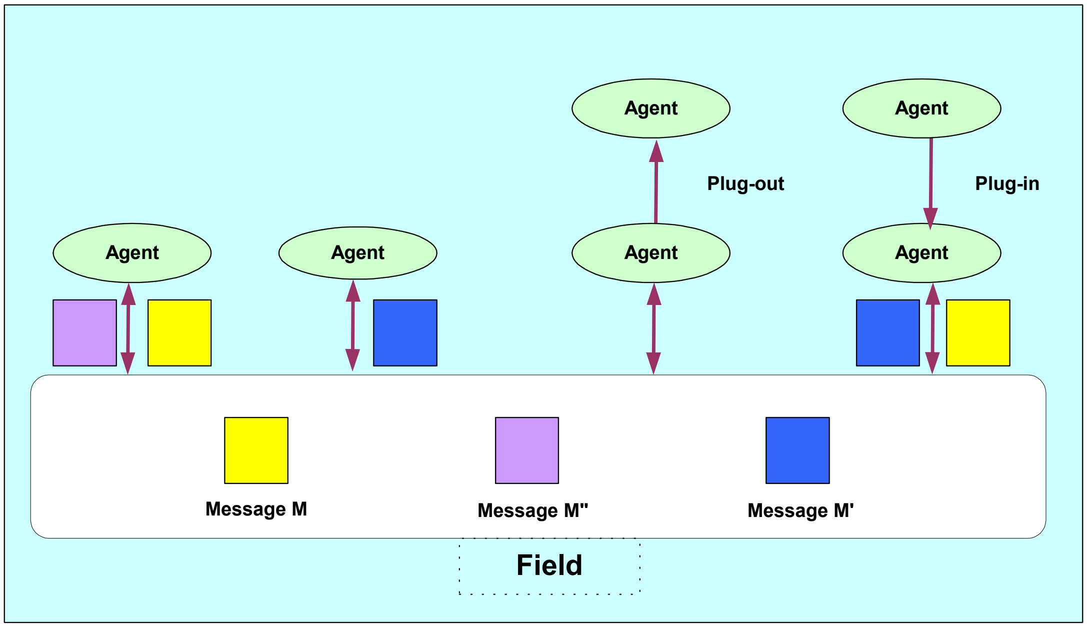

Figure 5-6: Inter-Agent Communication Using the Field [46]

5.5 Negotiation in Multi-Agent Based Systems

In multi-agent systems, it is necessary to allow software agents to negotiate with each other so they can cooperatively make decisions. In the negotiation paradigm, a client agent acts on behalf of the user and makes a proposal to a server agent. The server agent attempts to satisfy the client's proposal, which might involve communication with other servers. Having determined the available options, the server agent then offers the client agent one or more options that come closest to matching the original client agent proposal. The client agent may then request one of the options, propose further options, or reject the offer. If the server agent can satisfy the client agent request, it accepts the request; otherwise, it rejects the request [56].

To allow software agents to negotiate with each other, the following communication services are provided for client and server agents[58]. Client agents may do any of the following:

- **Propose service**. The client agent proposes a service to the server agent. This proposed service is negotiable, meaning that the client agent is willing to consider counter offers.
- **Request service**. The client agent requests a service from the server agent. This requested service is nonnegotiable**,** meaning that the client agent is not willing to consider counter offers.
- **Reject server offer**. The client agent rejects an offer made by the server agent.

Server agents may do any of the following:

- **Offer a service**. In response to client proposal, a server agent may offer a counter-proposal.
- **Reject client request/proposal**. The server agent rejects the client agent's proposed or requested service.
- **Accept client request/proposal**. The server agent accepts the client agent's proposed or requested service.

5.6 Integration Technologies for Evolutionary Agent-Based Systems

Agent-based systems can adapt and evolve with their environment and with changes in requirements. Two methods are presented – Wrapper Technology and XML.

5.6.1 Wrapper Technology

Many legacy applications cannot be easily integrated into a distributed agent-based framework. However, one approach is to develop wrapper objects. A wrapper object is a distributed application object that handles the communication and management of client requests to legacy applications [57]. A wrapper registers its service with the naming service so it can receive client service requests. In some cases, the legacy code is modified so the wrapper object can access it. However, this is often impractical because there is often little or no documentation and the original developers are no longer present. Consequently, wrapper objects often interface to legacy code through mechanisms such as files, which might be purely sequential or indexed sequential files. The wrapper object reads or update files maintained by the legacy application. If the legacy application uses a database, the database is accessed directly by using database wrapper objects that hide the details of how to access the database. For example, with a relational database, the database wrapper uses Structured Query Language (SQL) statements to access the database. Other options include making wrapper inputs appear as if they came from the keyboard and redirecting screen or printer outputs to the wrapper, a technique sometimes referred to as "screen scraping."

Legacy code can be converted by placing a wrapper around it and providing an agent interface to it [41]. The wrapper maps external requests from agents into calls in the legacy code. It also maps outputs from the legacy code into responses to the agent. By this means, agents can be used as an integration technology. As new requirements are added, the agent-based system will evolve by developing and incorporating new agents

5.6.2 XML Technology

XML is a technology that allows different systems to inter-operate through exchange of data and text[18]. XML is a markup language for documents containing structured information, as is the case with most documents. A markup language is a mechanism to identify structure in a document. The XML specification defines a standard way to add

---

[18] See: http://www.xml.com

markup to documents. An XML document contains both content (words, pictures, etc.) and tags to indicate how the content should be interpreted. For example, information content in a section heading has a different meaning from content in a textual paragraph or a database table. Standard XML documents can be used as a way of interchanging information between different intelligent agents that were not initially designed to inter-operate.

5.7 Component-Based Systems and Technologies for Agent-Based Systems

Component technology provides an environment for agent based systems to evolve more easily and to be more easily integrated with the multiple and diverse legacy systems, that support NS/EP requirements. A component is an active, self-contained object with a well-defined interface, capable of being used in different applications from that for which it was originally designed. In component-based systems, an infrastructure is provided that is specifically intended to accommodate preexisting components. Previously developed components are integrated with other components. For this to be possible, components must conform to a particular software architecture standard.

Middleware is a layer of software that sits above the heterogeneous operating system to provide a uniform platform above which distributed applications can run [59] (Figure 5-7). An early form of middleware was the remote procedure call (RPC). Other examples of middleware technology [60] are DCE (Distributed Computing Environment), which uses RPC technology, Java remote method invocation (RMI), Component Object Model (COM), Jini, Java Beans, and CORBA.

By providing a uniform way of interconnecting and reusing objects, middleware technologies such as CORBA, COM, and Java Beans promote component reuse, and hence are also referred to as component technologies [61]. Distributed component technologies include CORBA, which is an OMG standard, ActiveX and COM from Microsoft and Java Beans and Jini connection technology from Sun Microsystems.

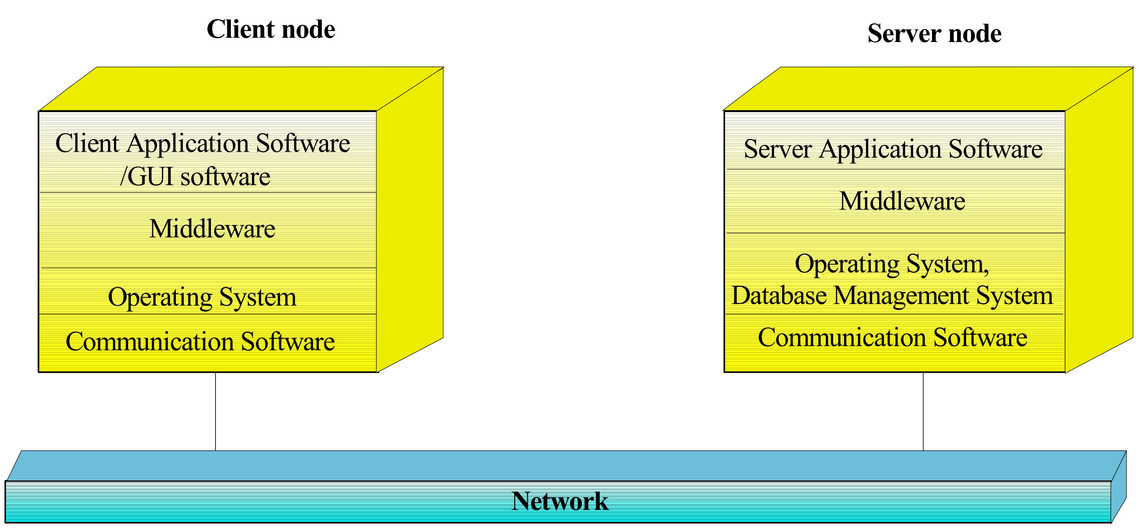

Figure 5-7: Example of Middleware in Distributed Component-based Applications

5.7.1 Common Object Request Broker Architecture (CORBA)

Object broker approaches such as CORBA [57] are designed to allow communication between objects on heterogeneous platforms. Server objects provide services that can be invoked from client objects by means of the Object Request Broker (ORB).

CORBA is an open systems standard developed by the Object Management Group, which allows communication between objects on heterogeneous platforms [57, 60]. The Object Request Broker (ORB) middleware allows client/server relationships between distributed objects. Server objects provide services that can be invoked from client objects by means of the ORB. In general, clients and servers are roles played by objects. Thus, an object might act as a client in a relationship with one object and act as a server in a relationship with a different object.

5.7.1.1 Object Request Broker

A client with an object reference can invoke any service (interface operation) supported by the server object via the ORB. The ORB provides the following [59]:

- **Location transparency.** The ORB locates the server object from the object reference.
- **Implementation transparency.** The client does not need to know how the server object is implemented, in what language or on what hardware it is running, or under what operating system.
- **Object execution state transparency.** If the server object is inactive, the ORB activates it before delivering the request to it.
- **Communication mechanism transparency.** The client is unaware of the underlying protocol used by the ORB.

5.7.1.2 CORBA Interface Definition Language

The server object's interface is written in an Interface Definition Language (IDL), which is programming language independent. The interface is defined separately from the implementation. The object's IDL specification is then translated into the target object's implementation language. The object's implementation is written directly in the target object's implementation language. OMG provides standard language mappings from IDL to the target language. Programming languages supported include C, C++, Ada 95, and Java. OMG IDL compilers generate client-side stubs and server-side skeletons. The client-side stub creates the request and sends it on behalf of the client. The server-side skeleton receives the request and delivers it to the CORBA object implementation. The CORBA stub and skeleton functionality are similar to RPC stubs and skeletons.

5.7.1.3 Static and Dynamic Binding

With static invocation, the stubs and skeletons are pre-linked with their executables. The static interfaces are defined by IDL definitions written by the developer. The IDL definitions are then compiled into stubs, skeletons, and header files as defined by the specific language mapping. This is relatively simple and efficient, but inflexible. This approach is shown in Figure 5-8.

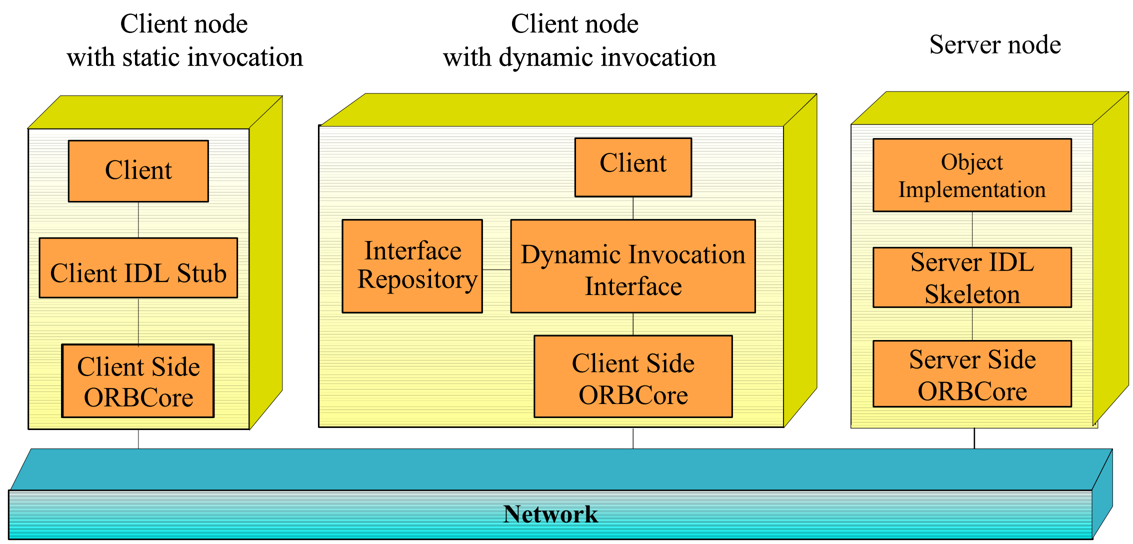

Figure 5-8: CORBA Architecture

A more flexible approach is dynamic invocation. With this approach, a client object can decide at runtime to which server object it should communicate. A server registers the IDL definitions in an interface repository, which can be queried dynamically at runtime. The client uses the Dynamic Invocation Interface, a generic stub, provided by the ORB and is independent of the IDL interface of the object being invoked. This approach allows a client to invoke an object at runtime without compile-time knowledge of its interface.

CORBA also supports a Dynamic Skeleton Interface at the server side, which provides a runtime binding mechanism for servers that need to handle incoming service calls for objects that do not have IDL-based compiled skeletons. This is useful for communicating with external objects such as gateways or browsers [57].

5.7.1.4 CORBA Services

The ORB allows a client to transparently invoke an operation on a server object. It provides a naming service. When a CORBA object is created, the object is given an object reference, which is a unique identifier. An object's reference may be obtained by means of a directory lookup. The client may then invoke the operation on the object. The naming service provides a directory service similar to the telephone white pages directory.

Another CORBA service is the trader service. This service allows object references to be retrieved based on matching characteristics associated with the object (such as type of service) to characteristics sent by the client. The trader service provides a directory service similar to the telephone yellow pages directory.

5.7.2 Component Object Model (COM)

Distributed Component Object Model (DCOM) is a Microsoft distributed object technology that builds on the Component Object Model (COM) architecture [62]. COM provides a framework for application interoperation in a Windows environment. DCOM allows a client to communicate with a component on a remote node. DCOM intercepts the client call and forwards it to the server. Although both COM and CORBA provide an IDL, CORBA is intended as a standard, whereas COM is proprietary and Windows-specific.

An ActiveX component is a software executable that conforms to the Microsoft COM standard and runs on Windows platforms. It can be loaded and executed by COM-compatible containers. Microsoft's Internet Explorer Web browser is enabled to handle ActiveX components.

5.7.3 Java Beans

Java Beans are a Java-based component technology intended for domain-specific applications [61, 63]. Java Beans are user interface client components, whereas Enterprise Java Beans are server-side components. A bean is a component that consists of a set of classes and resources. Beans can be assembled into applications by means of an assembly builder tool. During assembly, a bean's behavior can be inspected (referred to as "introspection") and can be customized. Beans can trigger events and handle incoming events. An assembly builder tool can determine the events a bean triggers and the events it receives. The assembly tool can then connect event source beans to event receiver beans. Customized and connected beans can be saved for later use.

5.7.4 Jini Connection Technology

Java Intelligent Network Infrastructure (Jini) is a connection technology for embedded systems and network-based computing applications whose objective is to make it easy to interconnect computers and devices. Jini is intended for digital devices such as cellular phones, digital cameras, televisions, and videocassette recorders. The Jini technology uses Java technology, and devices in a network are connected by using Java RMI.

Jini provides a lookup service, which performs a brokering role between service providers and clients [64]. Jini also provides protocols for discovery, join, and lookup. A provider of a service—for example, a digital videocassette recorder—registers with the Jini lookup service. Thus, a new provider must first dynamically locate the lookup service (referred to as "discovery") and then register itself with the lookup service (referred to as "join"). For each service it wishes to provide, the provider must upload a Java object that provides the interface for that service. A Jini client—for example, a digital camera—locates the Jini lookup service by using the discovery protocol. The client then uses the lookup service to locate an appropriate service—for example, a recording service provided by the videocassette recorder (VCR). The client then downloads a Java object from the lookup service, which allows the client to interact directly with the device. Thus, the digital camera would use the Jini lookup service to locate the VCR recording service, download the VCR recording Java object, and then communicate directly with the VCR.

5.8 DARPA Programs in Intelligent Information Integration and Agents

The Defense Advanced Research Projects Agency (DARPA) has sponsored several R&D programs. These programs have focused on integrating information from multiple heterogeneous sources and control of large communities of agents, organized into possibly heterogeneous agent-based systems.

5.8.1 *Intelligent Integration of Information*

DARPA funded research the Intelligent Integration of Information from 1992 to 1996, under the direction of Dr. Gio Wiederhold of Stanford University. A collection of papers on Intelligent Integration of Information, edited by Wiederhold [49] appear in a special issue of the Journal of Intelligent Information Systems. Here the problem is to access diverse data residing in multiple, autonomous, heterogeneous databases, and to integrate or fuse that data into coherent information that can be used by decision makers. To make the problem even more interesting:

- Data may by multimedia (video, images, text, and sound);
- Data sources may store data in diverse formats (flat files, network, relational-, or object-oriented databases);
- Data semantics data may conflict across multiple sources;
- Data may have diverse temporal and spatial granularities;
- Data that is interesting and valuable may reside outside the enterprise, in the open-source literature accessible via subscription services or on the World Wide Web (WWW) [65]; and
- Data may be of uncertain quality, and the reliability of the source may be questionable.

Clearly, one cannot expect to solve information integration and other large-scale system problems with a monolithic and integrated solution. Rather, the system should be composed of smaller components, with each component having the requisite knowledge to perform its tasks within the larger problem-solving framework. Thus, the use of Cooperative Intelligent Agents holds promise in helping to address, discuss and understand the issues in building next-generation intelligent information systems.

Bird [30] has proposed an agent taxonomy based on two client/server classes. They are Mobile Agents (Clients) for Content, Communications and Messaging Services and Static Agents (Servers). Bird notes that distributed intelligent systems share many of the same characteristics of multidatabase systems [66], in particular, distribution, heterogeneity, and autonomy. Knowledge and data would be distributed among various experts, knowledge bases and databases, respectively making problem solving a cooperative endeavor [67, 68].

There are several facets to the heterogeneity of information in systems [30]: syntactic, control and semantic.

- Syntactic heterogeneity refers to the myriad of knowledge representation formats [69], data definition formats to represent both knowledge and data.
- Control heterogeneity arises from the many reasoning mechanisms for intelligent systems including induction, deduction, analogy, case-based reasoning, etc. [51, 67, 68, 70-74].
- Semantic heterogeneity [75-77] arises from disagreement on the meaning, interpretation and intended use of related knowledge and data.

A third characteristic of intelligent systems is that of autonomy. There are several aspects to autonomy; in the control structure of an agent, in the extent to which an agent shares information with other agents [31, 51], the manner in which an agent associates with other agents, and structural autonomy in the way an agent fits into an organization of agents for problem-solving [32, 36, 37, 39, 74, 78-82].

Bowman et al [83, 84] describe a three-layer architecture for scalable Internet resource discovery, proposed by the Internet Research Task Group. Table 5-1 denotes the three-layer architecture that provides access to heterogeneous repositories, including those on the WWW.

Table 5-1: Three Layer Internet Information Architecture

Information Layer	Layer Service
Information Interface Layer	Users perceive the available information at this layer and may query and browse the data. This layer must support scalable organizing, browsing and search.
Information Management Layer	Responsible for the replication, integration, distribution, and caching of information.
Information Gathering Layer	Responsible for the collecting and correlating the information from many incomplete, inconsistent, and heterogeneous repositories.

The Intelligent Integration of Information Program identified a collection of services to support the following types of activities:

- Information Interface Layer. Thesaurus services, information search services, query decomposition and semantic query optimization services, and information presentation services;
- Information Management Layer. Information integration, real-time subscription and notification services, and information mediation services, e.g., knowledge rovers, facilitators, and brokers; and federation services;
- Information Gathering Layer. Wrapper services, query execution services, and data harvesting services.

The following are some services that are explained in more detail.

5.8.1.1 Thesaurus Services

The intelligent thesaurus [76] is an active data/knowledge dictionary capable of supporting multiple ontologies and allowing users to formulate and reformulate requests for objects. The intelligent thesaurus is similar to the thesaurus found in a library; it assists users in identifying similar, broader or narrower terms related to a particular search term, thereby increasing the likelihood of obtaining the desired information from the information repository. In addition, active rules may be associated with object types as well as their attributes and functions.

5.8.1.2 Federation Services

Federation services allow for a distributed client/server architecture [85] in which the constituent systems maintain authority and autonomy over their data which at the same time sharing certain information with the federation. Client software and/or server software is provided to members so that they can interface existing information systems with the federation [86, 87].

A Federation Interface Manager (FIM) is associated with each information system joining the federation. It is specialized into client and server FIMs. Each Client FIM consists of three subsystems, a Client Router, Client-to-Federal Translation Services, and Federal-to-Client Translation Services. The translation services map local data objects to the DPSC representation and vice-versa. The Client-FIM accepts local transactions (including query requests) from local clients in the format used by the local information system, and the client router determines whether the transaction is for a local server or a remote server of the federation. If the destination is a valid server, the request is passed to the Client-to-Federal Translation Services so the request may be translated to standard federal transaction format prior to routing it to the appropriate server.

The Server-FIM consists of Federal-to-Server Translation Services, Server-to-Federation Translation Services, and the Server Router. When a transaction arrives at a destination, the Federal-to-Server Translation Services translates the federal transaction into the local transaction format of the server. After translation the transaction is sent to the server router. The router receives all transactions, both locally and remotely generated, and logs them. Transactions are then queued for processing. Once the server has processed the transaction, the server router sends the response, if necessary, to the Server-to-Federal Translation Services, for server to DPSC translation, and then routes the response to the appropriate Client-FIM, where the response is translated into the local format.

5.8.1.3 Mediation Services

Mediation refers to a broad class of information integration services [48, 49, 88, 89]. One such mediation service is the integration of temporal data of differing granularity. This is of particular importance in the context of multidimensional databases and data warehousing applications, where historical data is integrated and analyzed for patterns and interesting properties.

A temporal mediator [90-92] consists of three components: 1) a repository of windowing functions and conversion functions, 2) a time unit thesaurus, and 3) a query interpreter. There are two types of windowing functions: the first associates time points to sets of object instances, and the other associates object instances to sets of time points. A conversion function transforms information in terms of one time unit into that in terms of some other time unit. The time unit thesaurus stores the knowledge about time units (e.g., names of time units and relationships among them). The time-unit thesaurus stores concepts such as the seasons, fiscal year definitions, and calendars, and effects translation of these time units into others.

Users pose queries using the windowing functions and desired time units using a temporal relational algebra. To answer such a user query, the query interpreter first employs the windowing functions together with the time unit thesaurus to access the temporal data from the underlying databases and then uses the time unit thesaurus to select suitable conversion functions, which convert the responses to the desired time units. Thus, a temporal mediator provides a simple, yet powerful, interface that supports multiple temporal representations in a federated environment. Temporal mediators may also be used to compare historical databases such as those needed for auditing and data warehousing purposes. Figure 5-9 shows a conceptual diagram of mediators working with application agents and resources agents to facilitate, broker, and integrate information and other resources.

5.8.2 Control of Agent Based Systems

The DARPA Control of Agent-Based Systems (CoABS) program has as its goal to enhance the dynamic connection and operation of military planning, command, execution, and combat support systems to quickly respond to a changing operational picture by building a framework for integrating diverse agent-based and legacy systems. One important aspect of this program is the CoABS Grid. The Grid [93] provides a scalable, agent-based interoperability infrastructure for distributed, heterogeneous, components - agents, objects, & legacy C2 applications. Its goal is to minimize programmer effort to integrate components and supports runtime component discovery and interoperability – especially among heterogeneous agent communities. The Grid uses

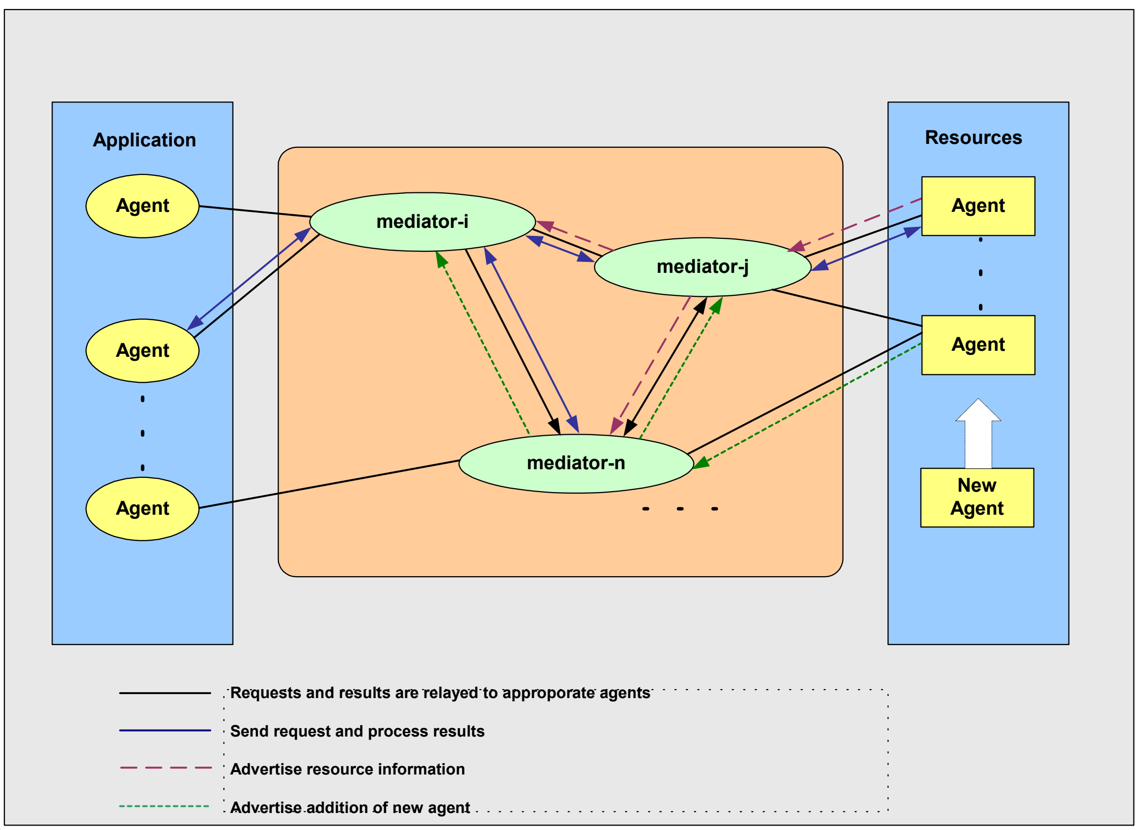

Figure 5-9: Mediators Acting as Intermediaries among Agents [46]

two key technologies. The first is software agent technologies, including multi-agent systems coordination and control, mobility, policy management, and reasoning technologies. The second is COTS Middleware Technologies, which includes open standards/software such as Java, Jini, and XML.The Grid provides a number of services that support component interoperability and component discovery for the building, testing and debugging of applications composed dynamically from heterogeneous components.

- Registration: enables components to make themselves known to others by providing information on their capabilities, needs and performance.
- Communication: provides messaging among components.
- Logging: captures messages and other activities by components.
- Visualization: shows components on the Grid and their interactions.
- Event Management: supports event subscription and notification.
- Security: provides encryption and authentication.
- Grid Management: supports the monitoring and control of Grid services
- Publish & Subscribe, and Multi-logging in future versions.

6 Intelligent Agent-Based Systems for Telecommunications

This section discusses the capabilities of intelligent agent-based systems for telecommunications, an area that has great potential for NS/EP as future networks emerge, which combine voice, data, and video, in which intelligent agents can play an important NS/EP role, in the area of service provisioning and taking proactive roles in NS/EP situations.

6.1 Capabilities of Agents for Telecommunications

Albayrak [94] states that it is useful to distinguish basic telecommunications applications from telematics services. A telecommunications application consists of the information system necessary to run, manage, and administer computer networks. Telematics unites information and communication technologies. A telematics service is a telecommunications service, such as call forwarding or caller id, which can be configured for the customer and for which the customer must pay. In agent-based telecommunications systems, the goal is to have a much more flexible configuration capability, where the customer can dynamically select the required services.

New telematics services have to be created quickly and combined from existing services. Services need to be dynamically adapted to customer needs. There are increasing requirements for quality of service, availability, and personalization. Decentralization of network management can be achieved using mobile, autonomous, cooperating agents. Albayrak concludes by stating that agent-oriented techniques (AOT) will lead to new frontiers in intelligent networks and active network management. For telecommunications applications, software agents must have specific capabilities in the areas of agent management, security, personalization, and mobility. Different types of agents are needed – manager agents for administration of the run-time platform, stationary agents as service providers, and mobile agents, which migrate from node to node in a distributed environment, for mobility support and service provisioning.

Telematics services have to meet several requirements because of rapidly changing markets. These include service personalization; support for user mobility; network convergence; on-demand service integration and combination; as well as security, manageability, transparency, robustness, and delegation.

6.2 Agent-Oriented Software Engineering for Telecommunications

Jennings [41] makes the case for Agent-Oriented Software Engineering (AOSE) for telecommunications, in other words to integrate intelligent agent concepts with software engineering concepts, which are important for building large scale systems that are well structured and maintainable. Using the Software Engineering concepts of decomposition, abstraction, and organization, he proposes a flexible, autonomous environment, involving multiple cooperating and negotiating agents, reflecting the decentralized nature of the problem. Jennings' concept is illustrated in Figure 6-2.

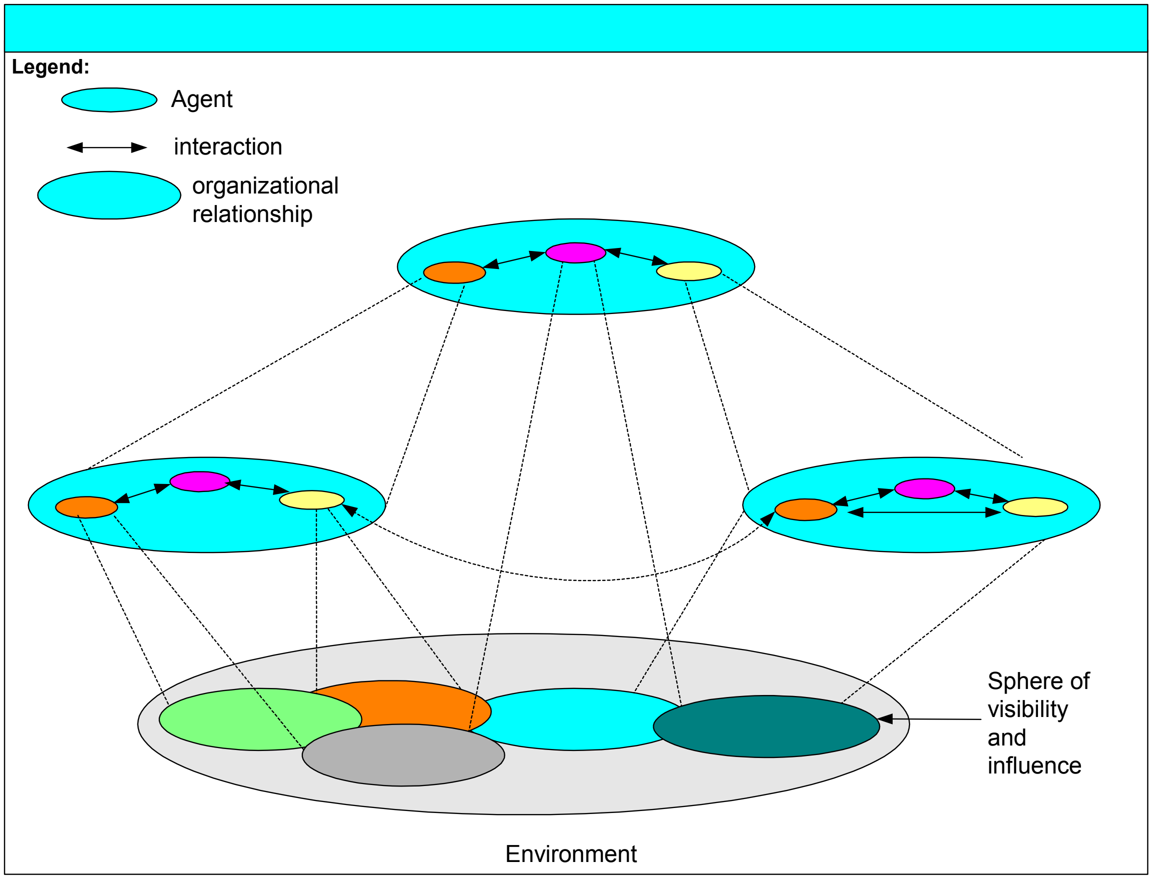

Figure 6-2: Canonical View of Multiagent System [41]

The information agents are distributed, autonomous, and cooperate with each other on behalf of their parties, which are users or organizations. The agents negotiate on behalf of their parties, making proposals, trading offers, making concessions, with the end goal of coming to an agreement. A case study is given in Section 6.6 of using this approach in intelligent agent-based negotiation for telecommunications.

6.3 Agent-based Telecommunications Services

Singh et. al.'s paper on agent-based telecommunications services [95] reviews the debate between the two main factions in the telecommunications industry:

- Traditional telephone companies advocate intelligent networks with increased intelligence embedded in network controlled by the telecom providers.
- Internet Service Providers (ISPs) prefer the "stupid network", which provides bit transport. Intelligence is distributed over the network and is not part of the network. End users choose the applications they prefer and invoke the services they desire without requiring changes in the network.

Singh advocates an approach where agents can perceive, reason, act, and communicate. Agents organize themselves into communities that help each other. Users can locate desirable services based on trustworthy, personalized, recommendations from peer agents. An end-user locates services directly or indirectly through recommendations made by peer agents. A response to an agent from a peer agent can be: no answer, bad answer, good answer, or referral to other agent(s). Each agent maintains weighting of service based on response received from the peer agent. Peer-to-peer systems allow users to exchange information directly without going through a server. This approach is illustrated in Figure 6-1 in which a query originates from agent A to agents B and C (message 1). A response is sent back from B (message 2) with referrals to agents D and E but not from C. Agent A then sends additional queries to agents D and E (message 3). Based on their responses (4, 5) agent A sends a query to agent F, as referred by E, (message 6) and gets a good response (message 7).

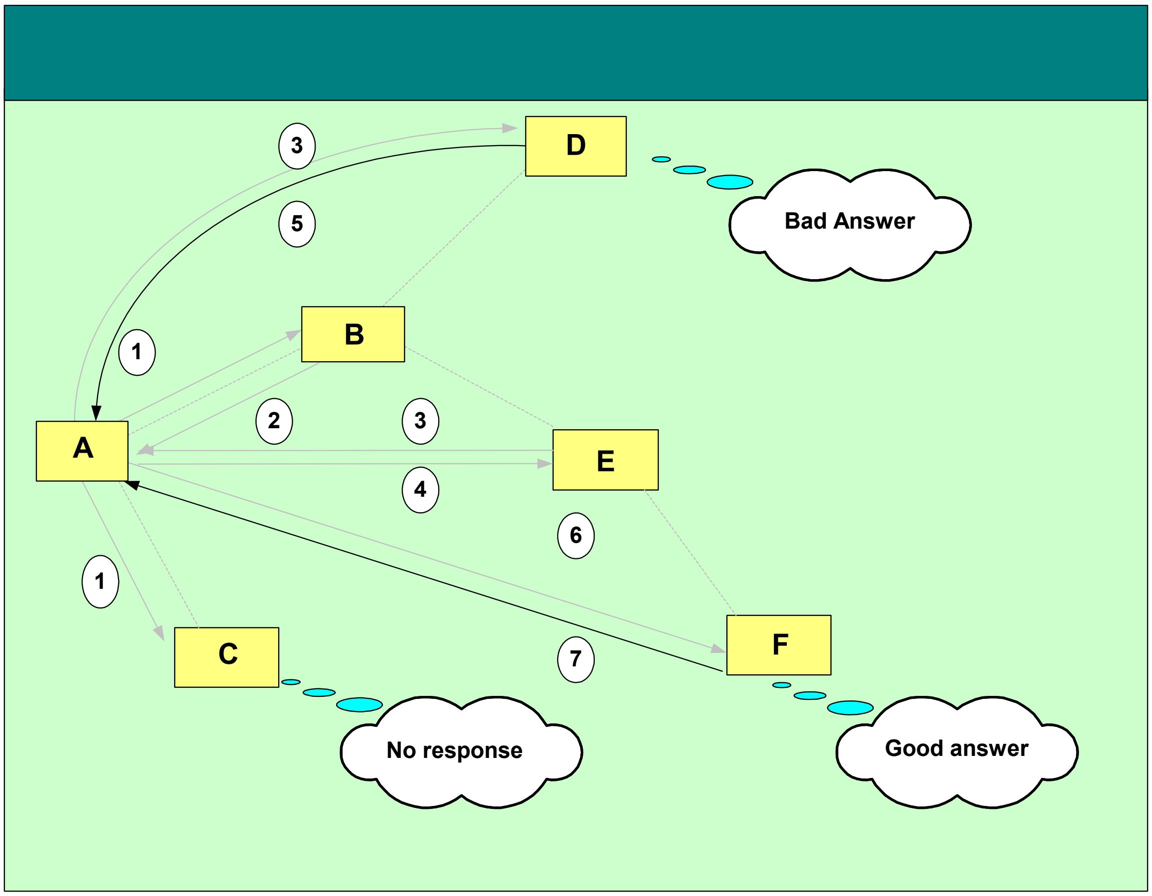

Figure 6-1: Agent-based Communication through Referrals [95]

6.4 Active Telecommunication Networks

This section reviews active telecommunications networks consisting of virtual active networks running intelligent network services on top of virtual private networks. Brunner and his colleagues [96] describe how to establish a framework that allows customers to run their own customized services over a provider's network. Active networking allows customized packet processing inside the network on a per-packet, per-flow, or per-service basis. From the management perspective, active networking allows rapid service deployment and flexible service management. A goal of active networking is to allow a user the flexibility of installing and running a network service as easily as installing a program on a PC. With this metaphor, the network, owned and operated by a service provider, is shared among several customers. However, for active networking technology to succeed, it is necessary to develop the concept of service creation and management. This would allow customers to install and run their own services on the network, while enabling service providers to effectively and efficiently isolate different customers from each other.

Brunner's paper describes a Virtual Active Network (VAN), which is a generalization of the Virtual Private Network (VPN) that captures the functionality and resources that a service provider offers to customers. The goal of a VAN is to provide customers with greater flexibility and control over the network services than a VPN. This is achieved by allowing customers not only the capability of running a network service (as with a VPN) but also to create service management functionality and manage their services without interacting the provider's management system.

Active networks transport active packets, which carry programs as well as data. These programs execute on network nodes, which could result in modifying the node's state and generating additional active packets to be sent over the network. In particular, the program in an active packet could modify or replace the node's packet processing function. For this to be possible, it is necessary to have active network nodes with higher throughput and a secure environment that allows different parties to share communication, CPU, and memory resources. Current research is addressing the problems of building high performance active network architectures, execution environments for active network nodes, integrating application specific functionality into the network layer, application specific network routing, and media scaling.

The paper proposes splitting the provisioning of an active network service into:

- Provisioning a generic service, which is provided with the cooperation of the provider.
- Installing a specific service, which is provided via the generic service interface without interacting further with the provider. The same interface is used for controlling and managing the service.

An example is given from Brunner's paper in Figure 6-3 of an active network with three VANs (Customer 1, Customer 2, and a Management VAN), and five nodes (nodes A-E) in a provider domain. In this example, Customer 1's VAN spans nodes A, B, C, and D. The service provider uses the Management VAN to provision and supervise the customer VANs. Each physical node has multiple virtual active nodes, which are also called Execution Environments (EEs). For example nodes A&D contain three virtual active nodes, Customer 1 EE, Customer 2 EE, and Management EE.

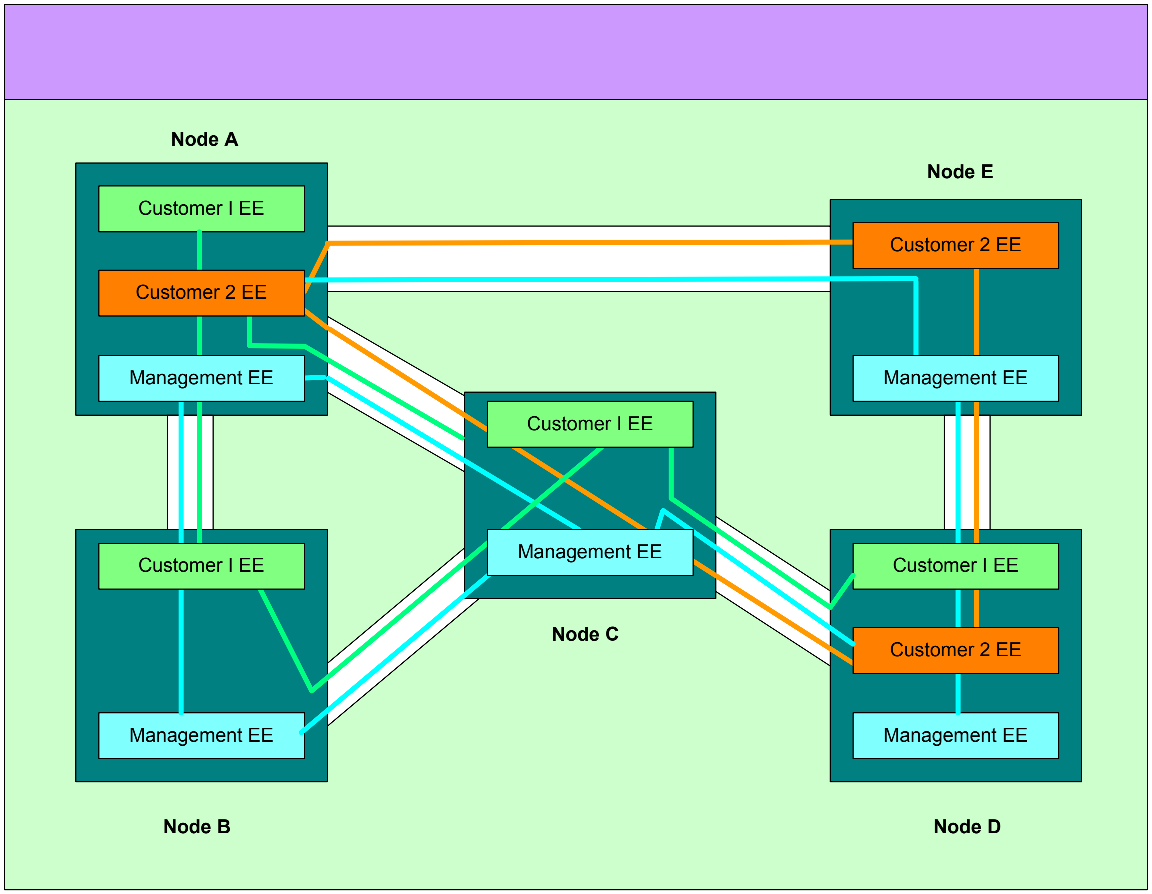

Figure 6-3: Virtual Active Networks [96]

6.5 Toolkits for Agent based Telecommunications Services and Applications

Several researchers have proposed toolkits [97] for realizing rapid development, deployment, and management of agent-based systems and services. Fricke et. al [97] propose a toolkit called Java Intelligent Agent Componentware (JIAC). With this toolkit, an agent is designed as a set of components managed by a component framework. The components within the agent interact by message passing. JIAC is oriented toward the ITU-T Recommendation X.701, entitled Information Technology – Open Systems Information – System Management Overview, which addresses technical characteristics of data transmission services. This standard specifies that management activities be divided into specific functional areas: fault management, configuration management, accounting management, and performance management.

The proposed agent architecture consists of a management framework and a security framework. Agents reside on platforms. The FIPA Platform Management Agent (MA) – provides basic functionality for the operating agents. The management activities consist of Directory Facilitator, Agent Management System, and Agent Communication Channel. A typical application consists of several platforms, which are loosely coupled by the mediation services provided by the Management Agents. The Management Agents control and supervise agent mobility between source and destination platforms.

There is also some management functionality within each agent addressing Fault, Configuration, Accounting (for e-commerce), and Performance (FCAP). The security framework provides security mechanisms for encryption and signing speech acts. The security protocol is provided for secure communication at the application layer, hence at a higher communication layer than Transmission Control Protocol / Internet Protocol (TCP/IP).

6.6 Negotiation among Intelligent Agents in Telecommunications Domain

Jennings presents a case study of intelligent agent-based negotiation [41] for provisioning a virtual private network, as illustrated in Figure 6-4. Consider a particular telecommunications application of intelligent agent-based negotiation to find the best videoconferencing service from among the services provided by various telecommunications service providers. This involves several users and organizations. A Personal Communication Agent (PCA) represents each end user. There are the service providers, which provide telecommunications services, e.g., for setting up the videoconference call. Each service provider is represented by a Service Provider Agent (SPA). There are network providers, which provide the network infrastructure upon which the services are provided. Each network provider is represented by a Network Provider Agent (NPA). First, the PCAs will negotiate among themselves on behalf of their users to find an appropriate time for the call. Next, one PCA communicates and negotiates with the videoconference SPAs. The negotiation is based on the cost and quality of the service. The winning SPA negotiates with the various NPAs to find the network provider that provides the best quality and bandwidth at the best price. Service provider and/or network provider agents may also negotiate among themselves in order to provide the desired service.

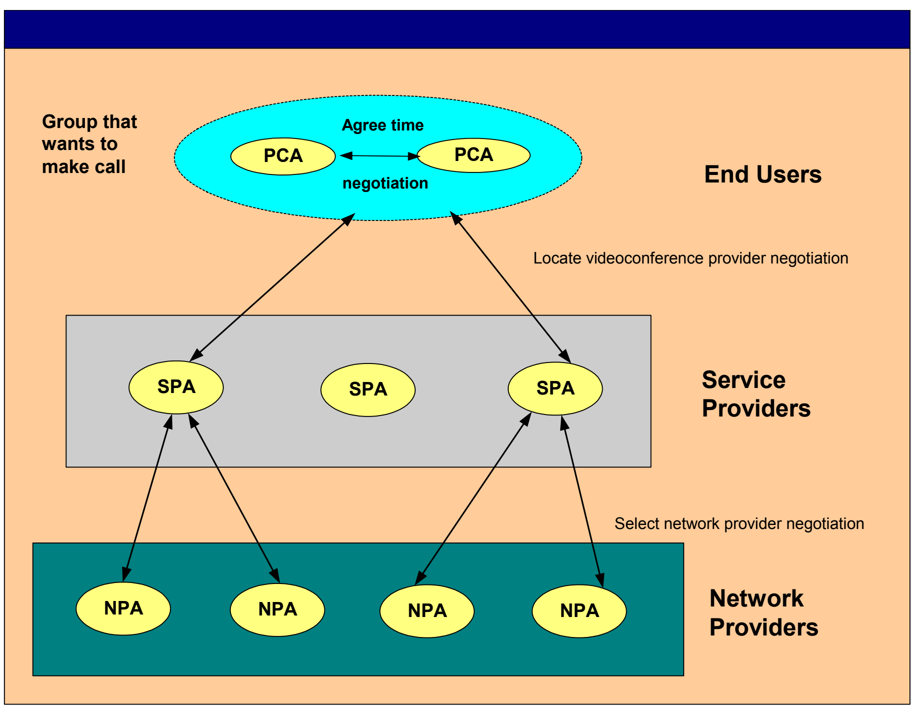

Figure 6-4: Dynamic Provisioning of Virtual Private Network through Agent Negotiation [41]

Consider the following detailed example involving negotiation among agents, which shows some of the negotiations described in the previous paragraph. The example addresses intelligent agent-based negotiation to find the best telecommunications service from among the services provided by various telecommunications service providers. In this example, the server agent is a videoconference agent who negotiates with various telecommunications service providers that provide a videoconference service. The negotiation protocol used is that described in Section 5.5 [56].

The videoconference agent [56] communicates with various service provider agents in order to find the most appropriate videoconference service. Assume that the client would like to set up a videoconference for a specific date, say December 10, 2001 at a specific time, 1-4 PM EDT, for a price of less than $700.

A collaboration diagram depicting the negotiation procedure is given in Figure 6-5 using the UML notation:

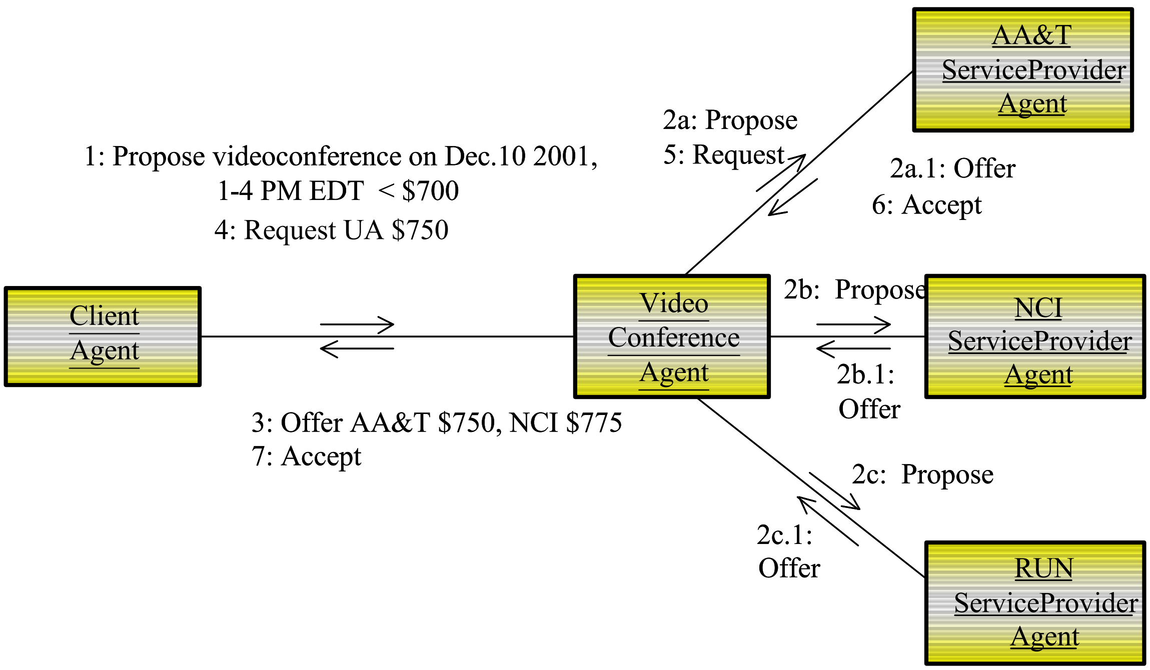

Figure 6-5: Example of Intelligent Agent Negotiation

1. Client Agent uses Propose Service to propose the videoconference to the Videoconference Agent with the constraints given above.
2. The Videoconference Agent proposes to three Service Provider Agents who provide a videoconferencing service, the AA&T Service Provider Agent, the NCI Service Provider Agent, and the RUN Service Provider Agent. It sends a Propose Service to the three respective Service Provider Agents for a videoconference on those dates. It receives offers from all three with the times and prices of the available slots for a videoconference.
3. The Videoconference Agent sends an Offer Service message to the Client Agent consisting of the available videoconference slots at the proposed price. If only more expensive videoconference slots are available, the Videoconference Agent

offers the cheapest it can find based on a metric combining low cost and Quality of Service. In this case, it determines that the two best offers for the proposed dates are from AA&T for $750 and NCI for $775. There are no videoconference slots below $700, so it offers the two available videoconference slots that come closest to the proposed price. It sends the Offer Service message to the Client Agent.

4. Client Agent displays the choice to the user, who selects the AA&T offer. The Client Agent may either Request Service (one of the choices offered by the Videoconference Agent) or Reject the server offer if the user does not like any of the options and propose a service on a different date. In this case, the Client Agent requests the AA&T videoconference slot.
5. The Videoconference Agent makes a reservation request to the AA&T Service Provider Agent.
6. Assuming the videoconference slot is still available, the AA&T Service Provider Agent accepts the reservation request.
7. The Videoconference Agent responds to the client agent's Request Service with an Accept Request message. If the videoconference slot were no longer available, the Videoconference Agent would send a Reject Request message.

7 Scenarios for Intelligent Agent-Based Systems and Smart Structures for NS/EP

7.1 Concept of Operations

This section investigates the integration of smart structures and smart sensors with intelligent agent-based information systems to provide a proactive role in Telecommunications Applications for National Security and Emergency Preparedness. It is assumed that network convergence is well established, Voice over IP is commonplace, and that all communication, voice, data, and video is over a packet switched network [1]. In comparison to the Network Convergence scenarios [1], which were reactive, many of these scenarios are proactive, taking NS/EP actions in anticipation of an impending catastrophe but before the catastrophe has actually struck. These scenarios also assume autonomous and cooperative intelligent agents patrolling the network, communicating with each other and with smart sensors and structures.

7.2 Hurricane scenario

Characteristics: Atmospheric disturbance. Downed telephone lines. High network traffic congestion. Need for hurricane warning messages. Advanced warning

With a hurricane situation, there are likely to be regional, local-level, and last-mile failures. There is the likelihood that telephone lines, carried on poles above the ground, would be down. Telephone lines carried underground would be more protected. In this situation, there would be restricted telephone access and also restricted access to the Internet by modem users using telephone lines to access the Internet.

In this scenario, smart structures and smart sensors are used for continuous monitoring and possible control of civil structures, such as bridges, roads, dams, and buildings for early detection of emergency situations, in this case flooding and hurricanes. Sensors in the atmosphere and at sea can be used for early detection of unusual weather situations that have a high probability of leading to a hurricane. As in the NASA JPL Sensor Web, the sensors can use wireless communications to communicate with local transmitters (for example on buoys at sea or weather balloons) or by satellite communications to the Internet, or by telephone through a gateway into the PSTN.

Intelligent agents would receive sensor data from multiple locations from where trends can be determined using intelligent information integration techniques, for example locality of the hurricane, speed of movement, and proactively send alarms to interested NS/EP parties. The intelligent agents would communicate among themselves to determine the extent of the damage. Agents would also be capable of sending alarms to each other and to NS/EP personnel using multicast technology and subscription/notification communication. They could reason about alternative NS/EP strategies and tailor the communication based on the appropriate actions.

As network outages occur and network traffic builds up, the intelligent agents can monitor the network and propose alternative routes for traffic. They can also automatically inform NS/EP personnel of congestion and outages. Through the use of smart sensors and smart structures, the intelligent agents can also inform NS/EP personnel of physical outages such as flooded roads or bridges that have been swept away.

Warning messages can be sent over the Internet. They can be routed to NS/EP personnel at computer workstations, laptops, digital phones (wired or wireless), and over cable TV. As pointed out in the Network Convergence report [1], there is a need for high priority traffic that can take precedence over regular Internet traffic.

7.3 Earthquake scenario

Characteristics: Downed telephone lines. Central offices out of action. High network traffic congestion. Major network traffic disruptions. Little or no advanced warning.

An earthquake is also likely to have regional, local-level, and last-mile failures. A severe earthquake is likely to be more disruptive than a hurricane as central offices as well as underground coax cables and telephone lines could be destroyed. Power failures and power surges could also cause widespread disruptions.

In this scenario, smart structures and smart sensors are used for continuous monitoring of civil structures, such as bridges, roads, dams, and buildings for early detection of emergency situations, in this case damage due to the occurrence of the earthquake.

Intelligent agents would receive the sensor data from multiple locations, as in the Sensor Web [15] from where trends could be determined, for example locality of the earthquake, extent of destruction, and proactively send alarms to interested NS/EP parties. As with the previous scenario, the intelligent agents would communicate among themselves to determine the extent of the damage. Agents would also be capable of sending alarms to each other and to NS/EP personnel using multicast technology and subscription/notification communication.

Robots can be used in situations that are hazardous for humans, providing this is considered cost-effective. In the earthquake scenario, special robots, controlled through haptic interfaces, could be sent to assess damage, remove hazardous materials, and locate trapped individuals.

As with the hurricane scenario, the intelligent agents can monitor the network for network outages and network traffic congestion, and propose alternative routes for network traffic. They can also automatically inform NS/EP personnel of congestion and outages. Through communication from smart sensors and smart structures, they can also inform NS/EP personnel of physical outages such as localities, roads, and bridges that have been destroyed by the earthquake. As with the hurricane scenario, warning messages can be sent over the Internet. They can be routed to NS/EP personnel at computer workstations, laptops, digital phones (wired or wireless), and over cable TV.

7.4 Terrorist bombing scenario

<u>Characteristics</u>: High network traffic congestion. Potential high casualties. Physical infrastructure damage. Little or no advanced warning.

In the terrorist bombing scenario, such as Oklahoma City or New York World Trade Center, the disruption is most likely due to local-level and last mile failures. There would be high network traffic congestion as well as some destruction of underground coax cables and telephone lines.

This scenario is more localized than the previous two scenarios. However, smart structures and smart sensors can detect the degree of damage in partially destroyed buildings, in particular for personnel who need to enter the building looking for injured personnel. If the smart structures or sensors were also destroyed, mobile robots could be dispatched into a building searching carrying smart sensors to assess the damage to the building and search for unexploded bombs. All communication from these smart sensors and mobile robots would likely to be via wireless communications to a local baystation and from there to the Internet.

7.5 Electromagnetic Pulse (EMP) Scenario

<u>Characteristics</u>: Electromagnetic Pulse shuts down central office. Some physical plant and equipment damage. High network traffic congestion. Major network traffic disruptions. This scenario considers a local EMP situation as might be caused by a terrorist organization. It does not consider a massive EMP situation caused by a nuclear explosion, which would be much more disruptive but is considered much less likely in the post-cold war environment. Little or no advanced warning.

The EMP scenario is likely to have an impact at the local and last-mile level. Thus having a central office out of action means that telephone access in the area covered by the central office would be lost. It is assumed that the area covered by the central is office is local and not regional.

This scenario is the most problematic, since it is likely that smart sensors and smart actuators would be disabled by the electromagnetic pulse. Intelligent agent based system could assess the extent of the damage and send emergency messages to NS/EP personnel. The intelligent agents would communicate among themselves to determine the extent of the damage. They would send alarms using multicast technology and subscription/notification communication.

8 Summary and Conclusions

This report has presented the state-of-the-art and state-of-the-practice in the areas of smart sensors and structures, intelligent systems, intelligent software agents, and telecommunications infrastructure to support agent-based monitoring, detection, planning, cooperation, and decision-making in support of NS/EP situations. Many of these technologies will be fully developed and will have significant impact on future telecommunications infrastructure. They will also have significant impact on NS/EP feature functionality and thus NCS must maintain close watch on individual technology developments. These technologies will have equal or greater impact on the Critical Infrastructure Program (CIP), now under definition, than on NS/EP. Specific recommendations relating to CIP should be addressed in a follow-on study. This report has focused on the evolving network infrastructure, wireless networks and Internet-based networks [1] to support both intelligent systems and smart structures. The major areas discussed were:

- **Extended Internet.** The extended Internet will help support the concept of smart sensors, smart structures, and agent-based systems. In the future, the number of Internet-aware devices – devices that can connect to the Internet and speak the Internet Protocol – will increase and their computing potential will allow them to perform intelligent tasks. The advances in next-generation internet-based devices and sensors will allow these sensors to communicate with intelligent agents that work cooperatively in solving NS/EP problems. With this much intelligence and knowledge in the distributed heterogeneous network, various types of devices, such as sensors, actuators, controllers, are likely to be organized into complex goal-driven systems that can be used for commerce, education, man-machine interactions, and NS/EP.

- **Transducers.** Transducer manufacturers seek to build low-cost, networked smart transducers. Transducers can be either sensors or actuators and are used in manufacturing, industrial control, automotive, aerospace, building, biomedicine, etc. The Technical Committee on Sensor Technology of the Institute of Electrical and Electronics Engineers (IEEE) Instrumentation and Measurement Society has sponsored a series of projects, called IEEE P1451, Smart Transducer Interface for Sensors and Actuators to interface smart transducers with diverse networks with and their associated protocols.

- **Sensor Web.** NASA, in conjunction with Cal Tech's Jet Propulsion Lab (JPL) is developing a web of sensors, called Sensor Web, which will be used to hunt for life on Mars. The Sensor Web consists of a collection of sensors contained in pods, which can be deployed over large areas. The smart sensors consume very little power, are able to communicate among themselves intelligently using advanced protocols, and connect to Internet-based controllers that can process the information.

- **Smart Structures.** The term "smart structures" refers to smart materials, intelligent materials, active materials, adaptive materials as well as actuators and sensors. The goal for smart structures and materials research is to incorporate them into existing materials, or augment existing structures with smart materials so as to control their behavior.

- **Intelligent Systems.** Intelligent systems originally emerged from the artificial intelligence community. However, in practice these early systems were often monolithic rule-based systems, and often became difficult to understand and maintain. More recently, a much more flexible, distributed and adaptable paradigm has emerged for intelligent systems, namely that of intelligent agent-based systems. An intelligent agent-based system is a highly distributed system, in which agents are active concurrent objects that act on behalf of users. Usually, several agents participate in the problem solving activity, communicating with each other. This leads to a more distributed and scaleable environment. Agents can be categorized in different ways, based on their mobility, based on their intelligence, or based on the roles they play in an agent-based system.

- **Agent-based systems**. Intelligent agents communicate with each other in many different ways, either directly or indirectly. For intelligent agent-based systems, the most flexible forms of communication are those that decouple agents from each other, in particular brokered communication and subscription / notification. With these approaches, agents can discover services provided by other agents, register the services they provide, and subscribe to services provided by other agents. In multi-agent systems, it is necessary to allow software agents to negotiate with each other so they can cooperatively make decisions.

- **Component technology**. Component technology provides an environment for agent based systems to evolve more easily and to be more easily integrated with legacy systems, an important consideration for NS/EP systems. Wrapper technology can also be used to integrate legacy applications into an agent-based system. XML is a technology that allows different agent-based systems to inter-operate through exchange of data and text.

- **Intelligent agent-based systems for telecommunications applications and services**. This is an area with potential for NS/EP as future networks emerge, which combine voice, data, and video, in which intelligent agents can play an important NS/EP role. For telecommunications applications, software agents must have specific capabilities in the areas of agent management, security, personalization, and mobility. Different types of agents are needed – manager agents for administration of the run-time platform, stationary agents as service providers, and mobile agents, which migrate from node to node in a distributed environment, for mobility support and service provisioning.

- **Potential Applications for NS/EP.** There is an opportunity for the integration of smart structures and smart sensors with intelligent agent-based information systems to provide a proactive role in Telecommunications Applications for National Security and Emergency Preparedness. Smart structures and smart sensors could be used for continuous monitoring of civil structures, such as bridges, roads, dams, and buildings for early detection of emergency situations such as earthquakes, flooding, and hurricanes. Sensors in the atmosphere and at sea can be used for early detection of unusual weather situations such as hurricanes. The sensors can use wireless communications to communicate with local transmitters (for example on buoys at sea or weather balloons) or by satellite communications to the Internet, or by telephone through a gateway into the PSTN. Intelligent agents would receive sensor data from multiple locations from where trends can be determined, for example locality of emergency, speed of movement, and proactively send alarms to interested parties.

It is apparent that smart systems and smart structures together with associated technologies will have a substantial impact on National Security and Emergency Preparedness telecommunications. In light of the above, specific observations with regards to NS/EP are as follows:

- Integrated intelligent systems incorporating smart structures will combine wireless and Internet technologies to allow devices to communicate over the Internet and PSTN networks.
- The Internet will evolve into the extended Internet, in which devices, such as cell phones, personal digital assistants, and sensors, will download programs to enable them to communicate with other devices.
- New network-based paradigms and architectures will need to be developed in order to manage the complexity of billions of devices using the extended Internet simultaneously,
- A promising new field is that of agent-based systems, in which autonomous intelligent agents (proxies for people and devices) will communicate and collaborate with each other to achieve goals.
- Agent communication protocols and languages will allow autonomous agents to discover, to negotiate, to broker, and to subscribe to services. The processing paradigm will be multicast and event-based and agents will be notified when events, to which they have subscribed, occur.
- The emerging consensus is that the intelligent network will provide value-added services that can be configured and managed by end users.
- The component-based technologies will allow for flexible applications that can be configured from reusable components, are capable of evolution, and can be more easily integrated with legacy systems.
- Wrapper technology will allow integration with legacy systems.
- XML will become the common document interchange language for devices and applications.
- Smart structures will be built using advanced material and control system technology so as to react and compensate for adverse situations such as earthquakes and hurricanes.

9 Recommendations

The National Communications System is exploring revolutionary concepts for the 21st century, in the areas of smart materials and structures, advanced network architectures, and intelligent systems. Each area is a specialized field with hundreds of scientists and engineers pursuing research and development in these diverse areas. While each of these areas should be monitored for new results that could impact NS/EP, true *synergy* will be obtained by integrating concepts from the various fields. It is recommended that the NCS should:

- Establish a study program to identify the existing and advanced NS/EP features, which should be included in the Extended Internet.
- Monitor the IEEE 1451 Committee on Sensor Technology for potential applications of transducers to the future telecommunications infrastructure.
- Monitor developments in sensor web arena. Sensor webs could be implemented nationwide to gather weather information, radiation fallout patterns, etc. Sensor web and component technologies, which interface legacy systems with agent-based systems, can be designed and implemented to process and understand a wide variety of sensor information.
- Keep abreast of other technologies such as smart structures, intelligent systems, and agent-based systems, which are all dependent on the availability of the Internet and its future evolution. Their application to NS/EP is yet to be determined but all have a strong potential impact on future telecommunications infrastructure developments. NCS should monitor the technology and standards developments as they relate to NS/EP and CIP.

Future architectures will enhance the NS/EP feature functionality, capabilities and responsiveness by using families of intelligent agents coupled with smart sensors and smart structures to provide active, intelligent, reasoning systems that:

- Respond to stimuli,
- Can react to extraordinary NS/EP situations,
- Are capable of planning, cooperating and taking timely actions, and
- Can learn and adapt over time.

The NCS should work with its governmental and industrial partners to promote NS/EP requirements in the specification, design and implementation of these integrated intelligent systems. NCS should endeavor to work with partners in fostering a testbed to create proof-of-concept experiments to explore the integration of these technologies in the context of NS/EP situations and scenarios.

Appendix A: Acronyms

ACL	Agent Communication Languages
ACM	Association for Computing Machinery
AMD	Active Mass Damper
AOSE	Agent-Oriented Software Engineering
AOT	Agent-Oriented Techniques
CACM	Communications of the Association of Computing Machinery
CIP	Critical Infrastructure Program
COBRA	Common Object Request Broker Architecture
COM	Component Object Model
COTS	Commercial Off-the-shelf
CoABS	Control of Agent-Based Systems
DARPA	Defense Advanced Research Projects Agency
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DEC	Distributed Computing Environment
EEs	Execution Environments
EMP	Electromagnetic Pulse
ENS	Enterprise Nervous System
ER	Electrorheological Fluids
FCAP	(For E-Commerce), and Performance
FIFO	First-in-first-out
FIM	Federation Interface Manager
FIPA	Foundation for Intelligent Physical Agents
GHOST	General Haptics Open Software Toolkit
HF	High Frequency
IC	Integrated Circuit
IDL	Interface Definition Language

IEEE	Institute of Electrical and Electronics Engineers
ISPs	Internet Service Providers
ITU-T	International Telecommunications Union
JIAC	Java Intelligent Agent Componentware
Jini	Java Intelligent Network Infrastructure
JPL	Jet Propulsion Lab
KQML	Knowledge Query and Manipulation Language
LAN	Local Area Network
MA	Management Agent
MEMS	Micro Electro Mechanical Systems
MR	Magnetorheological
NASA	National Aeronautics and Space Administration
NAWC/TSD	Naval Air Systems Division
NCAP	Network Capable Application Processor
NCS	National Communications System
NIST	National Institute for Standards and Technology
NPA	Network Provider Agent
NS/EP	National Security and Emergency Preparedness
OMG	Object Management Group
ORB	Object Request Broker
PCA	Personal Communication Agent
PCS	Personal Communications Service
PN	Public Network
PSTN	Public Switched Telephone Network
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SAC	Sensor and Actuator Center
SAMPLES	Sandia Agile MEMS Prototyping, Layout Tools, Education, and Services
SMA	Shape Memory Alloys
SPA	Service Provider Agent
SPIE	International Society for Optical Engineering

SQL	Structured Query Language
STIM	Smart Transducer Interface Module
TCP/IP	Transmission Control Protocol / Internet Protocol
TEDS	Transducer Electronic Data Sheet
UML	Unified Modeling Language
VAN	Virtual Active Nodes
VR	Videocassette Recorder
VE	Virtual Environment
VPN	Virtual Private Network
WWW	World Wide Web

Appendix B: References

[1] H. Gomaa and L. Kerschberg, "Network Convergence: Enabling Technologies for Advanced Telecommunications in support of National Security and Emergency Preparedness (NS/EP)," ComTek, Inc., Chantilly, VA, Technical Report August 1 2001.

[2] CTF, "Report of the Convergence Task Force," Washington, DC December 2000.

[3] NCS, "National Communications System: 1963-1998," National Communications System, Arlington, VA April 20, 1998.

[4] L. Kleinrock, "Breaking Loose," *Communications of the ACM*, vol. 44, 2001.

[5] V. Cerf, "Beyond the Post-PC Internet," *Communications of the ACM*, vol. 44, 2001.

[6] G. F. Colony, "My View: X Internet, Available at http://www.forrester.com/ER/Marketing/1,1503,214,FF.html," Forrester Research, 2000.

[7] M. J. Thompson, "The 'X' in What's Next," in *The Industry Standard, http://www.thestandard.com/article/0,1902,26675,00.html*, 2001.

[8] C. D. Howe, G. F. Colony, B. Doyle, C. Voce, and R. Shulman, "The X Internet," Forrester Research, Cambridge, MA, The Forrester Report May 2001.

[9] R. Schulte, "Designing the Agile Enterprise," presented at Application Integration: Moving Toward Total Business Integration, San Francisco, CA, 2001.

[10] K. Lee, "A Synopsis of the IEEE P1451 - Standards for Smart Transducer Communication, available from http://www.motion.aptd.nist.gov/," NIST, Gaithersburg, MD 1999.

[11] IEEE, "IEEE Std 1451.1-1999, Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model," Institute of Electrical and Electronics Engineers, Piscataway, NJ, Standard June 25 1999.

[12] IEEE, "IEEE Std 1451.2-1997, Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats," Institute of Electrical and Electronics Engineers, Piscataway, NJ, Standard September 26 1997.

[13] T. R. Licht, "The IEEE 1451.4 Proposed Standard," in *IEEE Instrumentation & Measurement Magazine*, vol. 4, 2001, pp. 12-18.

[14] M. Dunbar, "Plug-and-Play Sensors in Wireless Networks," in *IEEE Instrumentation & Measurement Magazine*, vol. 4, 2001, pp. 19-23.

[15] K. Delin, "JPL Sensor Webs Project; http://sensorwebs.jpl.nasa.gov/," JPL, 2001.

[16] A. Bridges, "NASA's Newest Search for Life Technology, http://www.space.com/businesstechnology/technology/web_sensor_000616.html," in *www.space.com*, June 16, 2000.

[17] C. Shakeri, M. N. Noori, and Z. Hou, "Smart Materials and Structures; A Review," presented at Fourth Materials Engineering Conference, Washington, DC, 1996.

[18] A. K. Jain and J. S. Sirkis, "Continuum Damage Mechanics in Piezoelectric Ceramics," in *Adaptive Structures and Composite Materials: Analysis and Application*, *ASME 1994 International Mechanical Engineering Congress and Exposition*, E. Garcia, H. Cudney, and A. Dasgupta, Eds. Chicago: ASME, 1994, pp. 47-58.

[19] A. R. Shahin, P. H. Mecki, and J. D. Jones, "Vibration Control Using Shape Memory Alloy Wires," in *Adaptive Structures and Composite Materials: Analysis and Application*, *ASME 1994 International Mechanical Engineering Congress and Exposition*, E. Garcia, H. Cudney, and A. Dasgupta, Eds. Chicago: ASME, 1994, pp. 47-58.

[20] Schonfeld, "Can Nanomuscles Replace the Low-Power Electric Motor? http://www.business2.com/articles/web/print/0,1650,16511,FF.html," in *Business 2.0*, 2001.

[21] J. M. Hollerbach, "Some Current Issues in Haptics Research," presented at IEEEIntl. Conf. Robotics and Automation, San Francisco, 2000.

[22] J. Zeidenberg, "International teaching centre for keyhole surgery opens in Hamilton, http://www.canhealth.com/dec99.html#anchor34332," in *www.canhealth.com*, 1999.

[23] BBC_Online, "Skull Surgery,

http://www.bbc.co.uk/science/tw/items/001213_skullsurgery.shtml," BBC Online, 2000.

[24] J. K. Salisbury and M. A. Srinivasan, "Phantom-Based Haptic Interaction with Virtual Objects," *IEEE Computer Graphics and Applications*, vol. 17, pp. 6-10, 1997.

[25] B. F. Spencer Jr. and M. K. Sain, "Controlling Buildings: A New Frontier in Feedback," in *IEEE Control Systems Magazine: Special Issue on Emerging Technology*, vol. 17, 1997, pp. 19-35.

[26] S. J. Dyke, B. F. Spencer Jr., M. K. Sain, and J. D. Carlson, "Modeling and Control of Magnetorheological Dampers for Seismic Response Reduction," *Smart Materials and Structures*, vol. 5, pp. 565-575, 1996.

[27] SANDIA, "Sandia National Labs MEMS Web Site; http://mems.sandia.gov/scripts/index.asp," 2001.

[28] D. Milojic, "Agent Systems and Applications," in *IEEE Concurrency*, vol. 8, 2000.

[29] N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes, "Hive: Distributed Agents for Networking Things," in *IEEE Concurrency*, vol. 8, 2000.

[30] S. D. Bird, "Toward a taxonomy of multi-agent systems," *International Journal of Man-Machine Studies*, vol. 39, pp. 689-704, 1993.

[31] M. Genesereth and S. P. Ketchpel, "Software Agents," *Communications of the ACM*, vol. 37, pp. 48-53, 1994.

[32] R. V. Guha and D. B. Lenat, "Enabling Agents to Work Together," *Communications of the ACM*, vol. 37, pp. 126-142, 1994.

[33] L. Kerschberg, "Knowledge Rovers: Cooperative Intelligent Agent Support for Enterprise Information Architectures," in *Cooperative Information Agents*, vol. 1202, *Lecture Notes in Artificial Intelligence*, P. Kandzia and M. Klusch, Eds. Berlin: Springer -Verlag, 1997, pp. 79-100.

[34] L. Kerschberg, "The Role of Intelligent Agents in Advanced Information Systems," in *Advances in Databases*, vol. 1271, *Lecture Notes in Computer Science*, C. Small, P. Douglas, R. Johnson, P. King, and N. Martin, Eds. London: Springer-Verlag, 1997, pp. 1-22.

[35] M. Klusch and L. Kerschberg, "Cooperative Information Agents IV: The Future of Information Agents in Cyberspace, Fourth International Workshop," in *Lecture Notes in Artificial Intelligence*, vol. 1860. Boston: Springer, 2000, pp. 283.

[36] M. Klusch and O. Shehory, "Coalition Formation Among Rational Information Agents," presented at Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-96), Eindhoven, Netherlands, 1996.

[37] M. Klusch and O. Shehory, "A Polynomial Kernel-Oriented Coalition Algorithm for Rational Information Agents," presented at Second International Conference on Multi-Agent Systems, Kyoto, Japan, 1996.

[38] P. Maes, "Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back," in *Special Issues of Robotics and Autonomous Systems*. Cambridge, MA, London, England: The MIT Press, 1990, pp. 194.

[39] D. Riecken, "M: An Architecture of Integrated Agents," *Communications of the ACM*, vol. 37, pp. 106-116, 1994.

[40] A. Haddadi, *Cooperation and Coordination in Agent Systems*. Berlin: Springer - Verlag, 1996.

[41] N. R. Jennings, "Agent-Based Approach for Building Complex Software Systems," *Communications of the ACM*, vol. 44, pp. 35-42, 2001.

[42] N. R. Jennings, "Coordination Techniques for Distributed Artificial Intelligence," in *Foundations of Distributed Artificial Intelligence*, M. P. Georgeff and N. R. Jennings, Eds.: Wiley InterScience, 1996.

[43] J. S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. Cambridge, MA: MIT Press, 1994.

[44] M. Wooldridge and N. R. Jennings, "Intelligent Agents: Theory and Practice," *The Knowledge Engineering Review*, vol. 10, pp. 115 - 152, 1995.

[45] M. Wooldridge, "Issues in Agent-Based Software Engineering," in *Cooperative Information Agents*, vol. 1202, *Lecture Notes in Artificial Intelligence*, P. Kandzia and M. Klusch, Eds. Kiel, Germany: Springer-Verlag, 1997, pp. 1-18.

[46] S. Case, N. Azarmi, M. Thint, and T. Ohtani, "Enhancing E-communities with Agent-Based Systems," *IEEE Computer*, vol. 34, pp. 64-69, 2001.

[47] L. Kerschberg and S. Banerjee, "An Agency-based Framework for Electronic Business," in *Cooperative Information Agents III*, vol. 1652, *Lecture Notes in Artificial Intelligence*, M. Klusch, O. Shehory, and G. Weiss, Eds. Berlin, et al.: Springer-Verlag, 1999, pp. 254-279.

[48] G. Wiederhold, "The Roles of Artificial Intelligence in Information Systems," *Journal of Intelligent Information Systems*, vol. 1, pp. 35-56, 1992.

[49] G. Wiederhold, "Foreword to Special Issue on the Intelligent Integration of Information," *Journal of Intelligent Information Systems*, vol. 6, 2/3, pp. 93-97, 1996.

[50] M. Wooldridge, "Semantic Issues in the Verification of Agent Communication Languages," *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 3, 2000.

[51] R. Fritzen, T. Finin, D. McKay, and R. McEntire, "KQML — A Language and Protocol for Knowledge and Information Exchange," presented at International Distributed Artificial Intelligence Workshop, Seattle, WA, 1994.

[52] Y. Labrou and T. Finin, "Semantics and Conversations for an Agent Communication Language," presented at IJCAI-97, Nagoya, Japan, 1997.

[53] J. Mayfield, Y. Labrou, and T. Finin, "Evaluating KQML as an Agent Communication Language," in *Intelligent Agents II*, vol. 1037, *Lecture Notes in Artificial Intelligence*, M. Wooldridge, J. Muller, and B. Nebel, Eds. Berlin: Springer-Verlag, 1992.

[54] FIPA_ACL, "Specification part 2 - Agent Communication Language," FIPA October 1997.

[55] H. Gomaa, *Designing Concurrent, Distributed, and Real-Time Applications with UML*. Reading, MA: Addison Wesley, 2000.

[56] H. Gomaa, "Inter-Agent Communication in Cooperative Information Agent-Based Systems," in *Cooperative Information Agents III*, vol. 1652, *Lecture Notes in Artificial Intelligence*, M. Klusch, O. Shehory, and G. Weiss, Eds. Berlin: Spring Verlag, 1999.

[57] T. Mowbray and W. Ruh, *Inside CORBA - Distributed Object Standards and Applications*. Reading, MA: Addison Wesley, 1997.

[58] J. Pitt, M. Anderton, and R. J. Cunningham, "Normalized Interactions between Autonomous Agents: A Case Study in Inter-Organizational Project Management," *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 5, pp. 201-222, 1996.

[59] J. Bacon, *Concurrent Systems*, Second ed. Reading, MA: Addison Wesley, 1997.

[60] R. Orfali, D. Harkey, and J. Edwards, *Essential Distributed Objects Survival Guide*: John Wiley and Sons, 1996.

[61] C. Szyperski, *Component Software*. Reading, MA: Addison Wesley, 1997.

[62] D. Box, *Essential COM*. Reading, MA: Addison Wesley, 1998.

[63] R. Orfali and D. Harkey, *Client/Server Programming with Java and CORBA*, Second ed: John Wiley and Sons, 1998.

[64] K. Arnold and others, *The Jini Specification*. Reading, MA: Addison Wesley, 1999.

[65] T. Berners-Lee, R. Cailliau, A. Loutonen, H. F. Nielsen, and A. Secret, "The World-Wide Web," *Communications of the ACM*, vol. 37, pp. 76—82, 1994.

[66] A. Sheth and J. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22, pp. 183-236, 1990.

[67] R. Davis and R. Smith, "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence*, vol. 20, pp. 63-109, 1983.

[68] C. Hewitt, "Open Information Systems Semantics for Distributed Artificial Intelligence," *Artificial Intelligence*, vol. 47, pp. 79-106, 1991.

[69] T. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, vol. 5, pp. 199-220, 1993.

[70] A. Bond and L. Gasser, "Readings in Distributed Artificial Intelligence." San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1988.

[71] R. Engelmore and T. Morgan, "Blackboard Systems," in *The Insight Series in Artificial Intelligence*, T. Morgan, Ed.: Addison-Wesley Publishing Company, 1988, pp. 602.

[72] L. Gasser and M. N. Huhns, "Distributed Artificial Intelligence," in *Research Notes in Artificial Intelligence*, vol. II. London and San Mateo: Pitman; Morgan Kaufmann Publishers, Inc., 1989, pp. 519.

[73] M. Tambe and P. S. Rosenbloom, "Event Tracking in a Dynamic Multi-Agent Environment," USC — Information Sciences Institute, Technical Report ISI-RR-393, September 11, 1994.

[74] M. Tambe, W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P. S. Rosenbloom, and K. Schwamb, "Intelligent Agents for Interactive Simulation Experiments," in *AI Magazine*, vol. 16, 1995, pp. 15-39.

[75] L. Kerschberg, "Knowledge Management in Heterogeneous Data Warehouse Environments," presented at International Conference on Data Warehousing and Knowledge Discovery, Munich, Germany, 2001.

[76] L. Kerschberg and D. Weishar, "Conceptual Models and Architectures for Advanced Information Systems," *Applied Intelligence*, vol. 13, pp. 149-164, 2000.

[77] J. P. Yoon and L. Kerschberg, "Semantic Update Optimization in Active Databases," presented at Proceedings IFIP WG2.6 Working Conference on Database Semantics (DS-6), Atlanta, 1995.

[78] M. Cutkosky, R. Englemore, R. Fikes, M. Genesereth, T. Gruber, W. S. Mark, J. Tenenbaum, and J. Weber, "PACT: An Experiment in Integrating Concurrent Engineering Systems," in *IEEE Computer*, vol. 26, 1993, pp. 28-37.

[79] T. Finin, "CIKM '94 Workshop on Intelligent Information Agents," T. Finin, Ed. Baltimore, MD, 1994.

[80] P. Maes, "Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back," in *Special Issues of Robotics and Autonomous Systems*. Cambridge, MA, London, England: The MIT Press, 1990, pp. 194.

[81] M. Klusch, "Using a Cooperative Agent System FCSI for a Context-Based Recognition of Interdatabase Dependencies," presented at Workshop on Intelligent Information Agents, CKIM Conference, Gaithersburg, MD, 1994.

[82] M. Klusch, "Utilitarian Coalition Formation Between Information Agents," in *Cooperative Knowledge Processing*, S. Kirn and G. O'Hare, Eds. London: Springer - Verlag, 1996.

[83] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, M. F. Schwartz, and D. P. Wessels, "Harvest: A Scalable, Customizable Discovery and Access System," University of Colorado, Boulder, Technical Report CU-CS-732-94, March 1995.

[84] C. M. Bowman, P. B. Danzig, U. Manber, and M. F. Schwartz, "Scalable Internet Resource Discovery: Research Problems and Approaches," *Communications of the ACM*, vol. 37, pp. 98—107, 1994.

[85] H. Gomaa and G. K. Farrukh, "An Approach for Generating Executable Distributed Applications from Reusable Software Architectures," presented at IEEE International Conference on Engineering of Complex Computer Systems, Montreal, Canada, 1996.

[86] H. Gomaa, D. Menascé, and L. Kerschberg, "A Software Architectural Design Method for Large-Scale Distributed Information Systems," *Journal of Distributed Systems Engineering*, 1996.

[87] L. Kerschberg, H. Gomaa, D. A. Menascé, and J. P. Yoon, "Data and Information Architectures for Large-Scale Distributed Data Intensive Information Systems," presented at Proc. of the Eighth IEEE International Conference on Scientific and Statistical Database Management, Stockholm, Sweden, 1996.

[88] G. Wiederhold, S. Jajodia, and W. Litwin, “Integrating Temporal Data in a Heterogeneous Environment,” in *Temporal Databases: Theory, Design, and Implementation*, A. U. Tansel, S. Jajodia, and others, Eds.: Benjamin/Cummings, 1993, pp. 563-579.

[89] G. Wiederhold, S. Jajodia, and W. Litwin, “Dealing with Granularity of Time in Temporal Databases,” in *Lecture Notes in Computer Science*, vol. 498, R. Anderson and others, Eds.: Springer-Verlag, 1991, pp. 124-140.

[90] X. S. Wang, S. Jajodia, and V. S. Subrahmanian, “Temporal Modules: An Approach toward Federated Temporal Databases,” *Information Sciences*, vol. 82, pp. 103-128, 1995.

[91] X. S. Wang, S. Jajodia, and V. S. Subrahmanian, “Temporal Modules: An Approach Toward Federated Temporal Databases,” presented at ACM SIGMOD International Conference on Management of Data, Washington, D.C., 1993.

[92] C. Bettini, X. S. Wang, E. Bertino, and S. Jajodia, “Semantic Assumptions and Query Evaluation in Temporal Databases,” presented at ACM SIGMOD International Conference on Management of Data, San Jose, CA, 1995.

[93] J. M. Bradshaw, N. Suri, A. J. Canas, K. Ford, R. Hoffman, and T. Reichherzer, “Terraforming Cyberspace,” in *IEEE Computer*, 2001, pp. 48-56.

[94] S. Albayrak, “Agent-Oriented Technology for Telecommunications,” *Communications of the ACM*, vol. 44, pp. 30-34, 2001.

[95] M. Singh, B. Yu, and M. Venkatraman, “Community-Based Service Location,” *Communications of the ACM*, vol. 44, pp. 49-54, 2001.

[96] M. Brunner, B. Plattner, and R. Stadler, “Service Creation and Management in Active Telecom Networks,” *Communications of the ACM*, vol. 44, pp. 55-62, 2001.

[97] S. Fricke, K. Bsufka, J. Keiser, T. Schmidt, R. Sesseler, and S. Albayrak, “Agent-Based Telematics Services and Telecom Applications,” *Communications of the ACM*, vol. 44, pp. 43-48, 2001.