Connection Oriented Routing Environment (CORE): A Generalized Device Interconnect

Ang-Chih Kao AKAO@MIT.EDU
Jorge Ortiz JORTIZ@MIT.EDU
Laboratory for Computer Science/MIT, 200 Technology Square, Cambridge, MA 02139 USA

Abstract

Inter-communicating devices are changing the way that services and connections are set up and established over a network. This paper introduces a new paradigm for setting up a peer connection between agents through a third party administrative agent called the connection oriented routing environment (CORE). In this new paradigm, the devices no longer follow the client/server model of communication, but are instead set up to be peers of each other. Through CORE, connections are established and maintained among devices. This paper will describe the CORE network system and will show how applications can be adjusted to fit the model CORE creates. It will also show that the new peer connection paradigm requires a re-structuring of current applications and also requires a new design strategy to accommodate the requirements such as authentication and peer specific data.

1. Introduction

Pervasive computing environments have emerged from an increase in the number of connected devices and from the improvement in performance of small devices. The ability to link multiple heterogeneous agents has introduced a new problem of complexity. Complexity is increased in the design of the network created and is inherent to the setup of heterogeneous interconnects. The client/server model is not always the correct solution for the connection of these devices. It is hard to tell which agent is the server and which agent is the client, and so deciding how to link them is unclear. This paper introduces a new paradigm, whereby each agent in the system is a peer of all the other agents and connections are established through a third party agent called CORE (connection oriented routing environment).

1.1 Third Party Connection Establishment and Agent Discovery

Part of the design of the third party arbiter paradigm consists of establishing connections between agents and maintaining resource availability information. In a home entertainment system, for example, the agents are the video displays, the speakers, the DVD player, the CD player, and any other components that are part of the system. CORE, in this scenario, is parallel to the human making the connections between the devices. The human is aware of the agents in the system and has wires to create links between appropriate devices. For example, the DVD player is connected to the video display, the CD player is connected to the speakers, and the speakers can also be connected to the video display or to the DVD player. Like the human arbiter, CORE establishes connections (links) between agents in the system. It would have to figure out which agent input and output streams are compatible by using a set of rules forwarded to it by the agents and processed by CORE.

1.2 Applications

In order to test current applications, we adjusted their architecture to fit that of third party management. We created three separate CORE dependent applications by rearranging their parts and connecting them using the third party paradigm. We created a voice controlled winamp application [1], a VNC application [2], and an RMI application [3] and noted the changes made to the system to get them to work successfully.

2. System Architecture and Design

When designing CORE the main focus was on handling links instead of having the end devices worry about what they can attach to. The CORE system consists of three parts: a discovery mechanism, a network infrastructure, and a rule processor. Each part is essential in the design and will be described in this section.

2.1 CORE Discovery Mechanism

First off, agents refer to all devices or services that wish to utilize CORE. All agents are made up of a pair of I/O streams that can be read or be written to at any time. The input streams and output streams can be connected to any other corresponding output and input streams, respectively. To discover the presence of a new service or device, an agent must inform CORE about itself, including its capabilities, when it establishes a connection with CORE.

2.2 CORE Network Infrastructure

CORE provides a network infrastructure for the connection among multiple agents. Each agent may connect to CORE using the local network and CORE may establish other links among connected agents. Referring back to the example of the home entertainment center and figure 1, you can see that even though the TV agent may connect directly to the VCR agent, it may also connect to CORE itself and CORE then provides the option to the TV agent of connecting to all available VCR agents or a remotely located VCR agent. This new functionality and set of options spawns new types of applications and creates new ways to provide services for users.

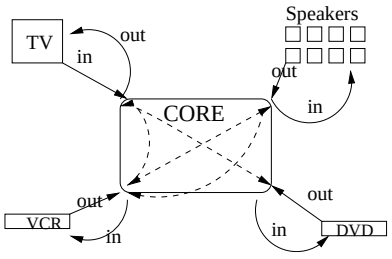

Figure 1. The figure above shows how the input and output ports are connected to CORE for each device and how CORE can be used in the home entertainment connection setup.

2.3 CORE Rule Processor

The rule processor is a programming interface very similar to a scripting language [4]. It gives the agents the option of creating links in a different way. There can be agents that manage how devices are connected and can change the way they are linked to introduce new applications. This makes it possible for the resource agents of CORE to function the same way and have the end devices function the same way, but for new services to be provided by switching links. All that CORE has to do is understand and process the rules so it can makes all this functionality possible.

3. CORE Applications- Results/Analysis

As previously mentioned, three applications were made to test how other applications can be changed to fit the CORE architecture. The first was the voice controlled winamp application; the second was the VNC (Virtual Network Computing) application, and last was the RMI (Remote Method Invocation) application.

The only application that did not work was VNC and RMI took significant tweaking to get it to work successfully. Voice controlled winamp functioned without a problem. Since there was no source for VNC available, not much could be done to find a solution to the problem.

Many applications were not built for the third party peer connection paradigm and so it takes some considerable effort to port applications over using CORE. We also noticed that there is still much left to look into when forming pervasive computing applications and there are many benefits to the paradigm presented.

4. Conclusion and Future Work

Computing devices and connectivity has become ubiquitous and will continue to grow in the coming future. With the number of devices with interconnectivity options, the need to explore new paradigms for their setup has become apparent. In this paper, we presented a third party arbiter that creates peer level connections among generalized devices. The CORE system shows a way to connect numerous devices and be able to manage and maintain connections of agents and resources as they become available. Right now, we have a working system of a generalize device interconnect, but still have many areas and features to explore. Some of those include access specification, whereby agents have the ability to restrict connection to other agents, and automatic service composition, which has CORE figure out to best way to link up it's agents to deliver a service. The Oxygen Research Group has started exploring some of these areas further and continues to study the benefits of this paradigm.

5. References

1. Nullsoft developer network. http://www.winamp.com/nsdn.
2. VNC Virtual Network Computing. http://www.uk.research.att.com/vnc/.
3. Sun Microsystems. *Using a Custom RMI Socket Factory, 2002*
4. Orlando Leon. An extensible communication oriented routing environment. Master's thesis, Massachusetts Institute of Technology, 2002.