Agents in Home Automation

Mike Rzucidlo

597B: Artificial Intelligence

July 30, 2001

Abstract

- In the next few years, home networking will become more common in every household. Agents will manage a specific home device and talk to other agents. They can find about their status, location or any other specific information related to that device. This can all be done by the interconnectivity of devices using a multi-agent architecture. Many companies are forming alliances to develop the next generation of home networking.

Overview

- Past
  - What has been researched and prototyped around the world?
- Present
  - What standards are being developed and refined?
- Future
  - How will the prototypes and standards developed today be used tomorrow?

Past

What has been researched and prototyped around the world?

What has been researched and prototyped around the world?

- The major researchers
  - Jae Chul Moon and Soon Ju Kang, School of Electronic and Electrical Engineering at Kyungpook National University in Korea
    - Proposed a multi-agent architecture for intelligent home network service
    - Priority Queue-Based IEEE1394 Device Driver Supporting Real-Time Characteristics
    - Real-Time Event Kernel Architecture for Home-Network Gateway Set-Top Box

Multi-agent architecture for intelligent home network service

- What is the problem?
  - Organizing distributed and heterogeneous home devices effectively
  - Supporting reliable communication between agent spaces

Introduction

- What could a typical home network consist of?
  - Digital cameras, TV's, Air Conditioners, VOD servers, etc..
- What are the problems connecting these devices together?
  - Adopting conventional centralized server-based intelligent software architecture is costly to maintain
  - Does not typically support security management, automatic power saving, intelligent room conditioning
  - Only performs low level device management and manipulation

Introduction

- How can these problems get solved?
  - Multi-agent based software architecture
  - Each agent in the home network covers some specific area
  - Each agent cooperates with each other to provide services
  - To support better communication, a multi-agent based middleware

Proposed Architecture Overview

- Home network consists of several agent spaces interconnected to each other
- Each agent manages either
  - a specific home device
  - a room-network
  - or several other agents using agent spaces and the space interconnect through which the status of all devices can be accessed regardless of the agent's location

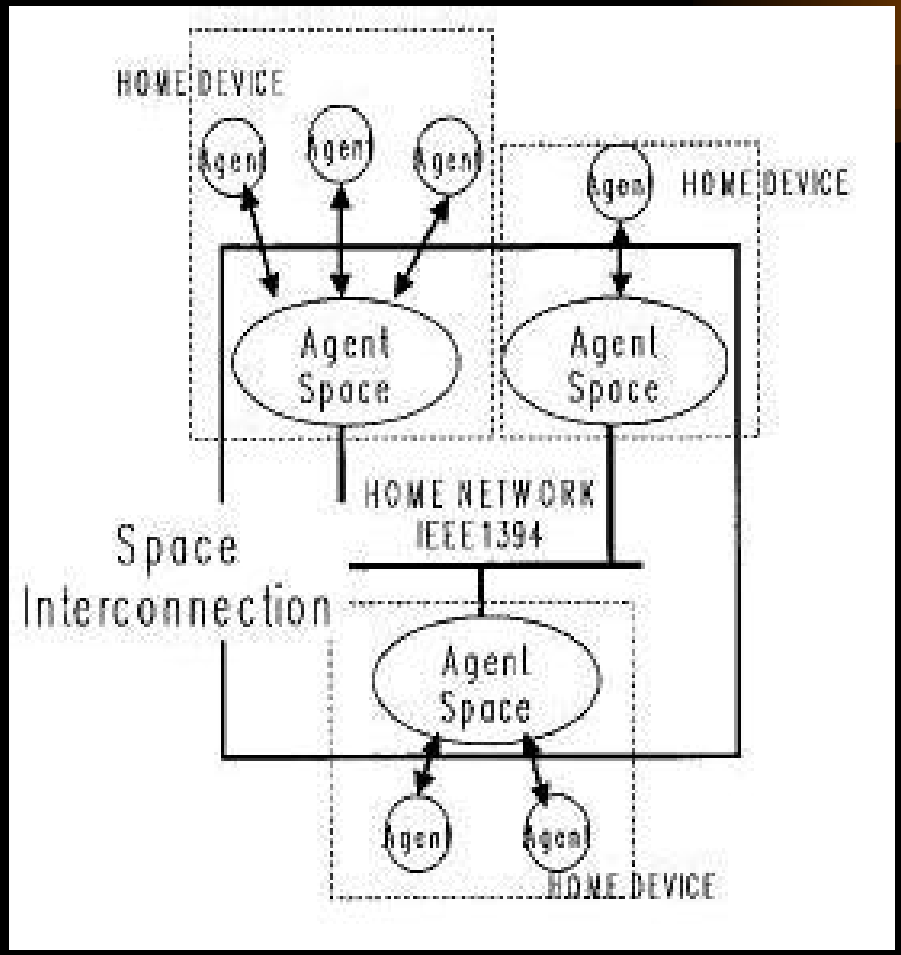
Proposed Architecture Overview

- A newly attached home device or agent can easily get information of another agent simply by reading or taking the information from the agent space
- An agent can announce its information to all other agents and devices by including it in the agent space
- Agents can provide a highly intelligent service since it is able to retrieve the information

Proposed Architecture Overview

- System has agents that are loosely coupled to each other
- The end result makes this architecture
  - Flexible
  - Scalable
  - Reliable

Proposed Architecture Overview



Agent Space based Multi-Agent Architecture

- Agent space consists of
  - Tuple space
  - Event kernel
  - Connection manager

Agent Space based Multi-Agent Architecture

- Tuple space
  - Acts as a temporal database that contains information generated by agents as a form of tuple, ie.
    - (“Light”, 2, “is”, true)
  - To retrieve information, an agent sends a query ie.
    - (“Light”, ?x, “is”, true)
  - When an event occurs
    - Tuple space notifies agent when newly arrived information is matched
    - Agent also registers even with a template

Agent Space based Multi-Agent Architecture

- Tuple space cont..
  - Each tuple space is replicated on each agent space
    - Agent can access device information anywhere transparently
    - Any operations that change the state of a tuple space are broadcasted to all agent spaces and synchronized
    - Agents can obtain status of home network or home itself through agent space not having to communicate with specific home devices

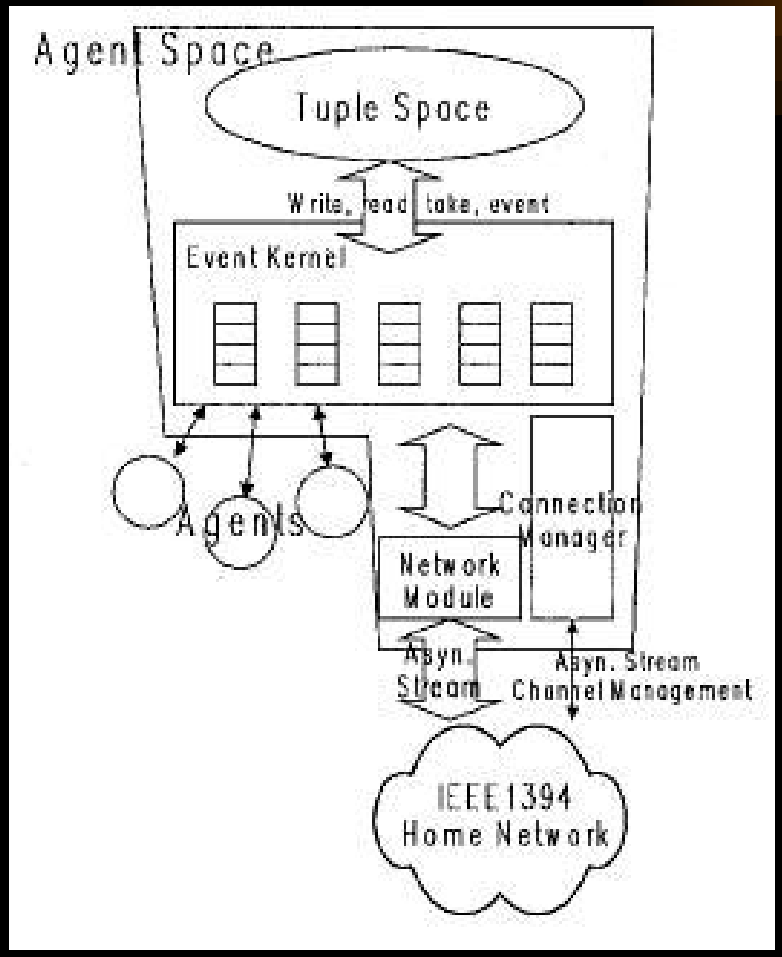
Agent Space based Multi-Agent Architecture

- Event kernel
  - Intermediates between the tuple space and the home network
  - Routes requests/responses to components of agents space, ie.
    - Tuple space
    - Network module
    - Agents
    - Connection manager
  - Schedules the requests based on priority
  - Network module performs actual network tasks

Agent Space based Multi-Agent Architecture

- Connection manager
  - Selects a channel management node if bus is initialized
  - Node allocates a channel number for broadcast to announce itself to all agent spaces
  - Agent spaces broadcast all operations using IEEE1394's asynchronous stream to synchronize the entire tuple space
  - Allocation of stream channel in IEEE1394 should be performed whenever network is initialized

Agent Space based Multi-Agent Architecture



Agent Space Structure

- Synchronizing all agent spaces need to be locked when inserting or deleting tuples
  - Decreases performance of agent space operation
  - Makes agent network unreliable because of failure of agent space will cause unreleased locking by non-response of the failure node

Agent Space Structure

- In proposed architecture
  - Most tuples have a valid period of time before it is deleted from tuple space
    - Room temperature is usually only valid for a few minutes
  - For reliable tuple operation, templates and tuples are marked as local or remote
    - Remote means that tuple is written by a remote agent space
      - For remote tuples, agent space does not remove them without acknowledgement from the source agent space
      - Agent space broadcasts request over the network and waits for acknowledgement
      - Source agent of the tuple then receives broadcasted request and removes from local tuple space
      - Source agent sends acknowledgement to requester
    - Local means that the tuple is written by a local agent space

Agent Space Structure

- In proposed architecture cont..
  - Newly attached agent space doesnot replicate entire tuple space of home network
    - causes excess network overload
    - needs a complex initialization protocol
  - New agent space has empty local tuple space
  - Remote and local write operations after initialization will insert tuples into the local tuple space

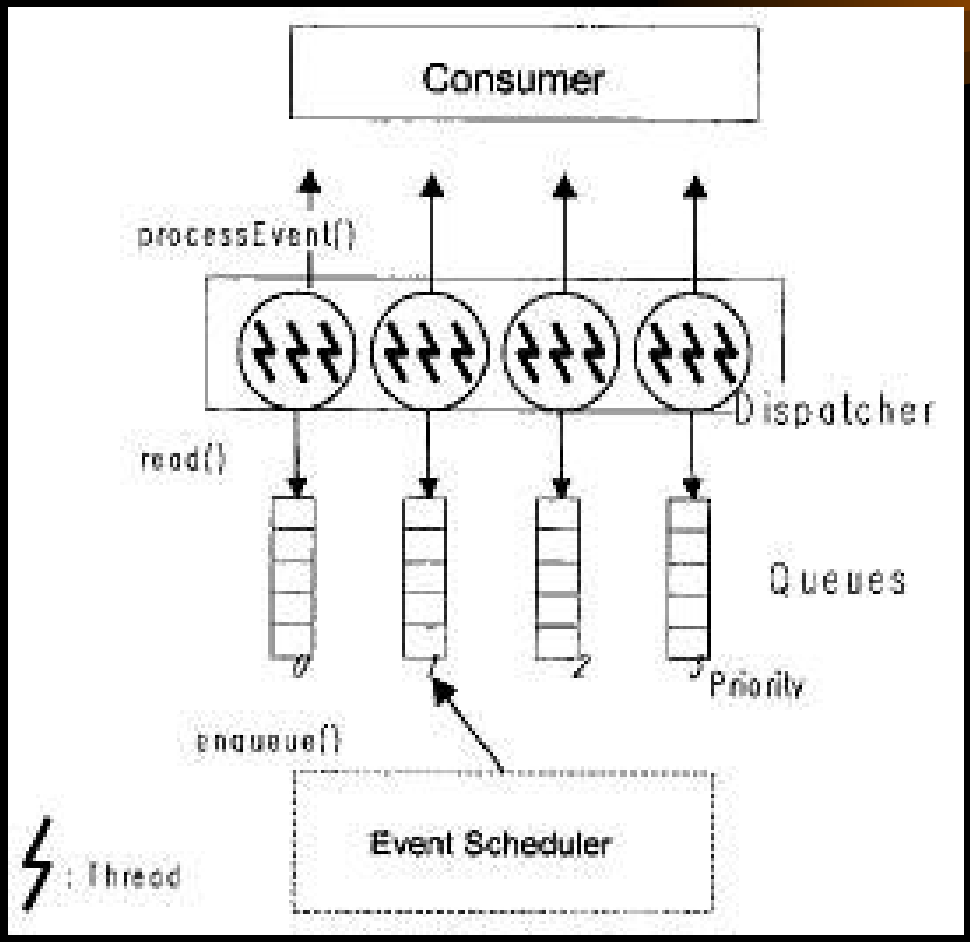
Agent Space Structure

- In proposed architecture cont..
  - Event kernel dispatches events to consumers within deadline of an event
  - To support this, event kernel includes
    - Event scheduler
    - Event dispatcher
    - Priority queues

Agent Space Structure

- Priority queues represent the priority of event
  - Events in higher priority queue preempt lower priority queues
- Event scheduler is added to preprocess an incoming event before priority queue
- Event-dispatching threads deliver events in a priority queue to event consumers (tuple space)
- Benefit of this architecture is that it produces fast and efficient preemption that is essential to real-time systems

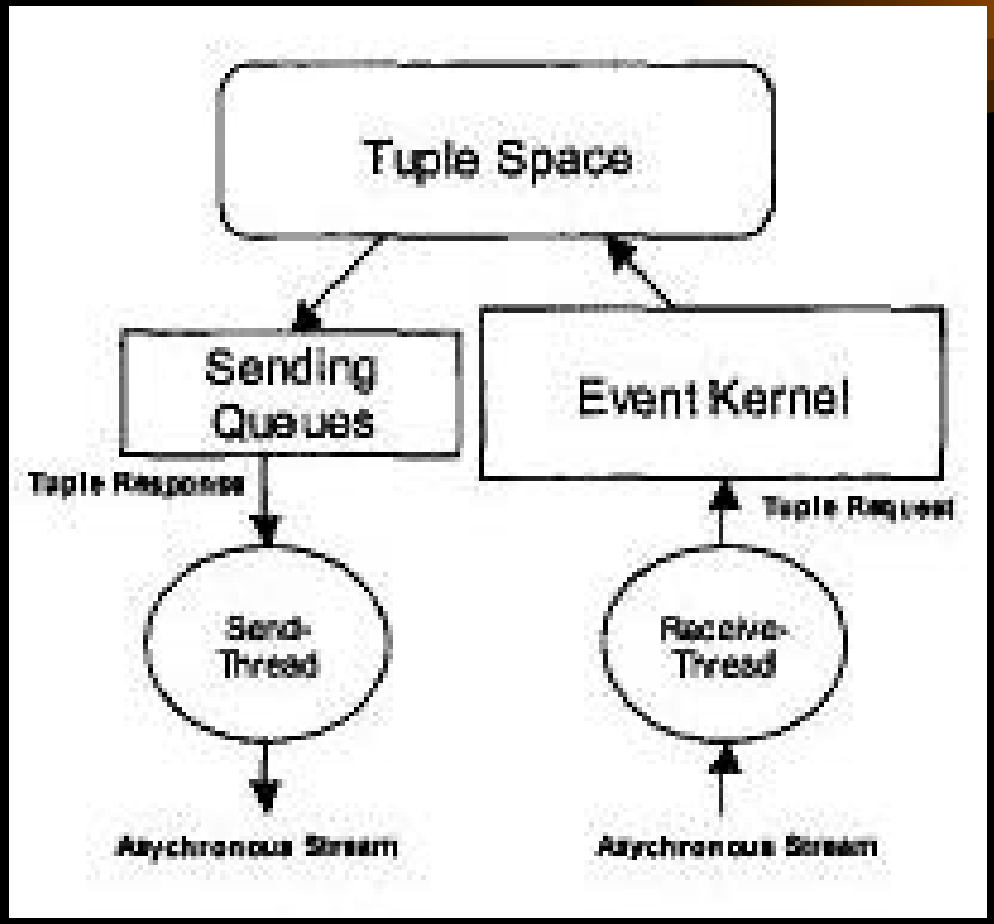
Agent Space Structure



Agent Communication Architecture

- Agent space is built based on asynchronous streams of IEEE1394 for agent communication
- Three asynchronous streams mapped to specific priority band of real-time OS
- Send-thread and receive-thread have highest priority of priority bands
  - Data from high priority band can preempt data from low priority band
  - After data reception, receive thread sends data to tuple space through kernel

Agent Communication Architecture



Implementation

- Hardware & Software used are
  - Embedded and minimum computing device
  - C/C++ in a real-time OS
  - UI is a Java applet
  - Test bed network consists of several single board computers (for intelligent agents) and a PC through an IEEE1394 network

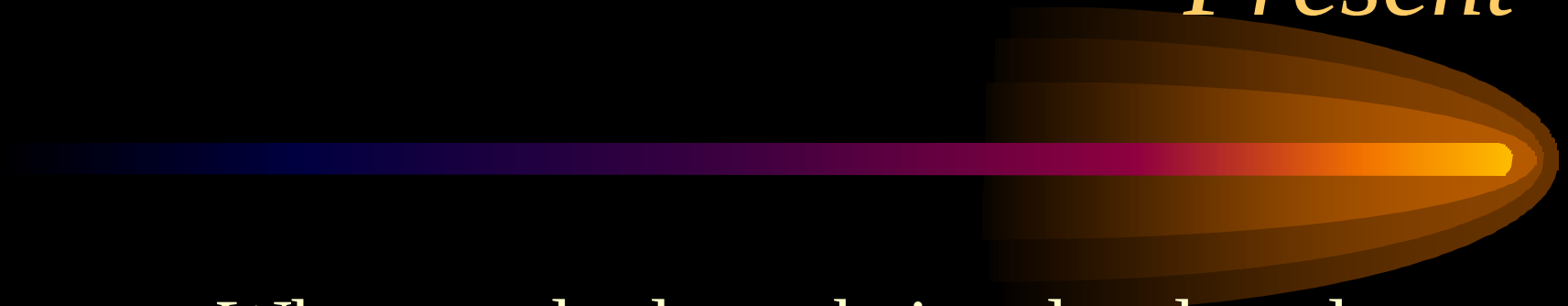
Implementation

- On testbed home network, security management was a test agent
- Security management services consist of one managing agent and several security devices
  - Security devices are emulated using software proxies
    - Each proxy generates security data and writes it into tuple space where the managing agent analyzes it to detect an intrusion
    - If intrusion is detected, the managing agent writes alarm message to tuple space
    - Several agents are waiting for alarm messages to alert through devices such as video, speaking, etc..

Implementation

- Security management agent was applied to different sets of security devices
- Security management agent could obtain goal for each set of security devices without any reconfiguring
- Result: architecture is flexible, reliable as proposed

Present



What standards are being developed and refined?

Currently developed standards

- Jini
- HAVi
- UPnP
- VHN

Jini

- Simple, Reliable, Scalable, Administration-free Operations
- Services carry code needed to use them
- Services are found and resolved using lookup services
- RMI-based communication
- Access Control List for each entity for security
- A grant of guaranteed access over a period of time

HAVi (Home Audio Video interop.)

- Multi-brand interoperable
- Open platform, architecture neutral
- Any device can interact with and control any other device
  - Distributed control, no single master
- Devices can present their User Interface on other devices
- A/V streams transport in high quality and real-time
- Extensible, “Future-Proof” open architecture
- Peer-to-peer architecture
- Downloadable upgrades
- Support for legacy devices

UPnP

- Enabling discovery and control of devices
- Designed to support zero-configuration
  - A device can dynamically join a network
  - Obtain an IP address
  - Convey its capabilities
  - And learn about the presence and capabilities of other devices—all automatically
- Automatic discovery for a breadth of device categories from a wide range of vendors
- Uses standard TCP/IP and Internet protocols

VHN

- Device Discovery uses Home Network Directory Page, Some HTML files
- User-to-Device Model uses Web based control model(HTML and HTTP)
- Service Registration and Discovery uses Interface Repository
- Device-to-Device Model and Event Handling uses XML based Remote Procedure Call

Future

How will the prototypes and standards developed today be used tomorrow?

In the news…

- Linux set-top API's planned
  - Founding members of TV Linux Alliance are ATI Technologies, Broadcom, STMicroelectronics, Motorola, Pace Micro Technology, Liberate, Lineo and Sun Microsystems
  - Alliance hopes that set of API's
    - Will determine device drivers
    - Make it easier to support device drivers
    - And ease of porting middleware to OS
  - Alliance also hopes to release Linux API specs by years end and make reference solutions available by 2002

In the news…

- Java in home networking
  - Major focus on how to deliver an efficient Java run-time system on embedded devices
  - Little work has been carried out to ensure that Java offers the type of functionality needed to control consumer devices that operate in the home
    - HAVi fills this need
  - CE companies reviewing HAVi as a full interoperability specification for consumer devices
  - Manufacturers weighing implications of full compliance (which adds memory and processing burden to devices)
  - Trade-off is implement full compliance or incrementally

Conclusion

- Many years have passed already and many different protocols developed
- The standards and protocols that will survive will be open-ended (think PC hardware…)
- The products that will cash-in, will be the companies that have the
  - Best Development tools,
  - Least politics (is this possible?),
  - Product with ease of use (installation, security, usability, etc.)
- Again, it’s Linux vs. WinCE

References

- Jae Chul Moon and Soon Ju Kang, “A Multi-Agent Architecture For Intelligent Home Network Service Using Tuple Space Model”, IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, August 2000, pp.791-4.
- Emil M. Petriu, Nicolas D. Georganas, Dorina C. Petriu, Dimitrios Makrakis and Voicu Z. Groza, “Sensor-Based Information Appliances”, IEEE Instrumentation and Measurement Magazine, December 2000, pp. 31-35.

References

- Junko Yoshida, “Linux set-top APIs planned”, EE Times, June 18, 2001. http://content.techweb.com/se/directlink.cgi?EET20010618S0024
- Junko Yoshida, “Alliance backs common Linux API for set-tops”, EE Times, June 13, 2001. http://www.eetimes.com/story/industry/systems_and_software_news/OEG20010613S0076
- Roger Lea, “Java pours into home network realm”, EE Times, June 14, 2001. http://www.eetimes.com/story/OEG20010614S0036

References

- Charles J. Murray, “Interface eases Bluetooth into everyday products”, EE Times, June 6, 2001. http://www.eetimes.com/story/OEG20010606S0076
- Hyo-Sang Lim, Dong-Hwan Park, Soon-Ju Kang and Bong-Jin Oh, “Priority Queue-Based IEEE1394 Device Driver Supporting Real-Time Characteristics”, IEEE Transactions on Consumer Electronics, Vol. 46, No.3, August 2000, pp. 825-33.

References

- Rodger Lean, Simon Gibbs, Alec Dara-Abrams and Edward Eytchison, “Networking Home Entertainment Devices with HAVi”, Computer Magazine, September 2000, pp. 35-43.
- Jae-Chul Moon, Hyo-Sang Lim and Soon-Ju Kang, “Real-Time Event Kernel Architecture for Home-Network Gateway Set-top-box (HNGS)”.

References

- Peter M. Corcoran, “Mapping Home-Network Appliances To TCP/IP Sockets Using A Three-Tiered Home Gateway Architecture”, IEEE Transactions on Consumer Electronics, Vol. 44, No. 3, August 1998, pp 729-36.
- Peter M. Corcoran, “Browser-Style Interfaces To A Home Automation Network”, IEEE Transactions on Consumer Electronics, Vol. 43, No. 4, November 1997, pp. 1063-9