VPcogma: A Light-Weight Cooperative Middleware for Ubiquitous Embedded Devices

Nobuo Kawaguchi[1]

[1] Information Technology Center, Nagoya University
1, Furo-cho, Chikusa-ku, Nagoya , 464-8601 AICHI, Japan
kawaguti@nagoya-u.jp
http://www.cogma.org/en/index_overview.html

Abstract. This paper introduces a light-weight middleware for embedded devices named "VPcogma". VPcogma is based on Virtual Processor and Code Translation technology. For the sake of the code translation, same binary code can run on different CPUs and can move across networks with dynamic state. It also enables to run with less memory and less CPU powers for satisfying the requirements for embedded devices. The design of VPcogma is based on our java based mobile-agent middleware called "cogma". Simple communication feature of cogma is also implemented on VPcogma. We also introduce the code translation system from Java-based cogma into C-based VPcogma. One can easily develop a mobile software on java-based rich system, and then translate it into light-weight system.

1 Introduction

The ubiquitous information environment is becoming true, where various information services can be used easily, always, and anywhere. By information technology and device technology in recent years, various devices are connected to information networks and various services have spread rapidly. In order to integrate various network information devices and services, a universal middleware for various types of devices (embedded, portable, appliance, etc.) is essential, which supports software development for a ubiquitous information environment. We describe the requirements for the universal middleware for building a ubiquitous information environment, and presents our middleware named “cogma”[6] which satisfies them. We also introduce our test bed environment called “cogma room” in which we have installed various kinds of sensors and appliances using cogma.

. Development of “cogma room” let us know that most of networked devices do not require “Java-based” rich engine. So we have decided to develop a light-weight middleware named “VPcogma”. We employ the running-environment “intent” for the dynamic deployment of software on different architecture. Programming of VPcogma is based on C and VPcogma also has a same architecture with cogma. So we have developed a code translator from java into C to keep compatibility.

This paper is composed as follows. In the next section, we introduce our Java based middleware "cogma" and its feature. Then we describe about our test-bed named "cogma room" in section 3. Section 4 introduces VPcogma and translator.

2 "cogma" A Universal Middleware for Ad-hoc Environment

For the middleware of ubiquitous information environment, we should take into consideration the following requirements to design a universal middleware for ubiquitous information environment. 1) Dynamic change, addition, and deletion of the devices, 2) Cooperative use of devices, 3) Integration of heterogeneous devices and heterogeneous network environment, 4) Minimization of the operation by the user, and 5) Minimization of the environmental load by information devices. Considering these requirements, we have developed a middleware named cogma (Cooperative Gadget for Mobile Appliances). Cogma has the following features.

(A) Lightweight middleware.
Small footprint implementation based on Personal Java/J2ME and intent.

(B) Dynamic code/state transfer.
Dynamic movement of code/state of software between nodes is possible while the other software components (named "codget") are running.

(C) Simultaneous use of two or more different type of network-link.
Cogma can use heterogeneous communication devices, such as TCP/IP, Serial, HTTP, etc.

(D) Autonomous discovery mechanism of other nodes.
TCP/IP link utilize Hello Packet to discover the other nodes on the same network.

(E) Simplicity of management and communication mechanism of mobile software.
Easy to learn how to develop a mobile software.

3 "cogma room" A Test Bed for Universal Middleware

In order to exemplify the usefulness of the middleware, we have constructed a smart space as a test bed named "cogma room"(Fig. 1). In cogma room, various sensors, such as temperature, humidity, human detection, brightness, and power monitor, are introduced. Temperature sensors are installed in 15 places, and can gain the temperature distribution in the 50-square meter room. Brightness of the 8 ceiling lights can be remotely controlled. The networked screen, projector, PDP, X10 devices, and the matrix switcher are also remotely controllable. Cogma room is designed on the assumption that it should be a live office to which people perform research activities daily in the real world. So, most of devices can be controlled from the user's note-PC or PDA. In this smart space, different kinds of networks, such as LonWorks (Echelon), 2-wire network (Matsushita Electric Industrial), and MicroLan(MAXIM), are simply integrated by cogma.

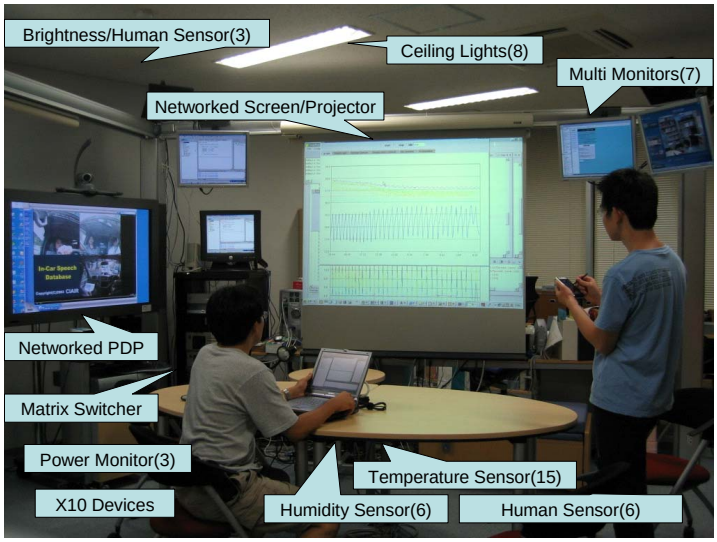


Fig. 1. "cogma room" a smart space controlled by cogma

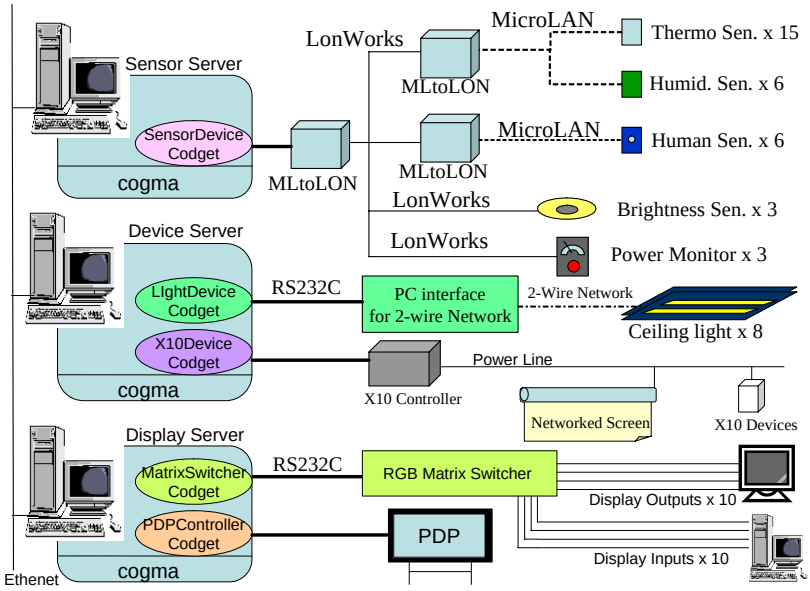


Fig. 2. Overview of components in "cogma room"

3 VPcogma : Light weight implementation of cogma

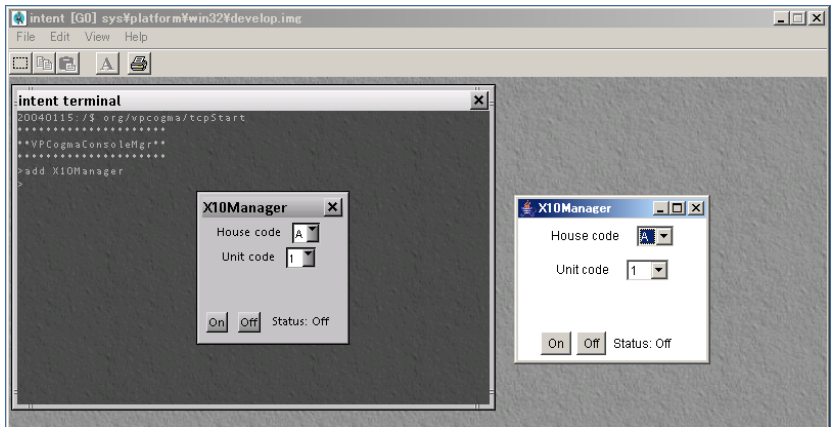
From the experience of constructing a smart-room, we have noticed that cost-requirements in java based cogma cannot be ignorable. For example, it is not suitable to use a java-based brightness sensor or java-based remote switch. Because, usually java requires some memory and CPU power. But also, it is required to satisfy the features for the devices in the ubiquitous information environment described in Section 2. So we decide to develop a light-weight middleware which still have a dynamic feature like cogma. We employ “intent” from TAO for running environment of VPcogma. Intent runs almost every CPU and it enables that binary code can be transferred over different CPUs like a byte code on Java. So we implement “cogma architecture” on intent using C language. It results a light-weight middleware named “VPcogma”. “VP” means “Virtual Processor” in intent. Table 1 shows memory requirements of VPcogma in Win32 environment.

Table 1: Memory requirements of VPcogma (Win 32 Environment)

	ROM Image	RAM Requirements
Serial Communication	650kbyte	100kbyte
TCP/IP Communication	880kbyte	800kbte
Full Feature with GUI	1.9Mbyte	2.3Mbyte

4.1 Translator of Java into VP-C

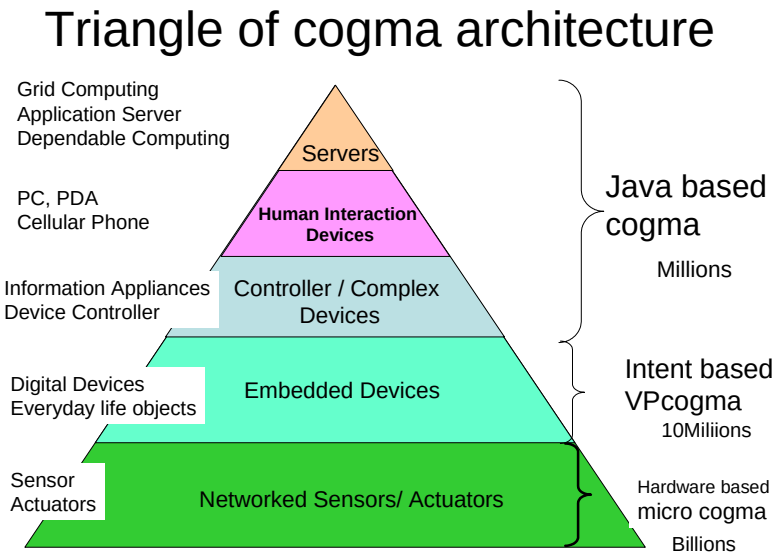
We have developed a translator from Java codget into VP-codget in C. This enables that a quick development of codget in Java and then translate it into C for deployment. We also implement compatible libraries of Java AWT and classes. Figure 3. shows codgets running on cogma and VPcogma.



.**Fig. 3.** Translated codget (left) in VPcogma and original codget(right) in Java.

5 Our Vision

We are currently considering the idea for developing smaller middleware. VPcogma is OK for embedded devices. But it is still too big for sensors and actuators. So we need to develop a simple mobile middleware named “micro cogma”. Our vision is to develop a hardware based middleware for dynamic configuration of the system.



.**Fig. 4.** Anatomy of devices for ubiquitous environment with cogma architecture.

7 Conclusion

In this paper, we briefly introduce our experience with cogma, cogma room, and VPcogma. We also describe about our vision for future. The cogma architecture is exemplified to work well in ubiquitous environment in our test-bed. So we need to make it more simple for sensors and actuators.

References

1. Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S., EasyLiving: Technologies for Intelligent Environments, Handheld and Ubiquitous Computing(HUC), pp. 12-27..(2000).

2. Brad Johanson, Armando Fox, and Terry Winograd, The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms, IEEE Pervasive Computing 1:2, pp.67-75. (2002).
3. H. Morikawa, M. Minami, and T. Aoyama: STONE: Context-aware Network Service Architecture, IEICE Technical Report IN2001-12, May 2001.IN2001-12,(2001).
4. Okoshi, T., Wakayama, S., Sugita, Y., Iwamoto, T., Nakazawa, J., Nagata, T., Furusaka, D., Iwai, M., Kusumoto, A., Harashima, N., Yura, J., Nishio, N., Tobe, Y., Ikeda, Y. and Tokuda, H.: Smart Space Laboratory Project: Toward the Next Generation Computing Environment, IEEE Third Workshop on Networked Appliances (IWNA 2001), (2001).
5. Nobuo Kawaguchi, Katsuhiko Toyama, and Yasuyoshi Inagaki, MAGNET: Ad-Hoc Network System based on Mobile Agents, Special Issue for Mobile Agents for Telecommunication Applications, Computer Communication, Vol.23, pp.761--768(2000).
6. Nobuo Kawaguchi: Cogma: A Middleware for Cooperative Smart Appliances for Ad hoc Environment, Proc. of International Conference on Mobile Computing and Ubiquitous Networking (ICMU2004), pp.146-151 (2004).
7. cogma Project home page: http://www.cogma.org/