

DISCUSSION ON CSCW METHODS FOR EMBEDDED SYSTEMS

Tao Yu¹, Tan Liu¹, Shuzhen Yang², Wenbin Wang¹

¹*CIMS & Robot Center of Shanghai University, Shanghai University, China; Email: alanltnew@citiz.net.* ²*Electromechanical Engineering Institute, Shanghai Second Polytechnic University, Shanghai, China.*

Abstract: This paper discusses the Computer Supported Collaborative Work (CSCW) methods that can facilitate the work of embedded systems. The focus is on how to design and build effective embedded network architecture, and which CSCW tools should be used in embedded system. A brief conception of CSCW and embedded system is presented, followed by designing and building an embedded network system currently available, and CSCW tools used in embedded systems are discussed additionally. Finally, predictions are made on what CSCW for embedded systems might expect to be in future years.

Key words: CSCW, Embedded System.

1. INTRODUCTION

Doing business over Palms is cheap and convenient, and a commercial conference can be hold just through Palm communications. C4P (CAD, CAPP, CAM, CAE and PDM) [1] can be collaboratively worked in the microcontrollers such as SCM or Soc (System on chip). A project can be carried out from general engineer to partial managers, and to every executor without a desktop computer. A work-flow can be finished smoothly through your GSM. You can control robots or machine tools just with your mobile phone through the internet, and so on.

All of above are embedded products in our daily lives. The CSCW research for general computers has been presented for a long history, but the CSCW study for embedded systems is done with even less people. For the reason that embedded systems are too many and they are designed for all kinds of different special purposes, the research is complicated to carry out.

This research work is financially supported by the Science and Technology Development Foundation of Shanghai Science and Technology Committee (036505008 and 05DZ11001).

Please use the following format when citing this chapter:

Yu, Tao, Liu, Tan, Yang, Shuzhen, Wang, Wenbin, 2006, in International Federation for Information Processing (IFIP), Volume 207, Knowledge Enterprise: Intelligent Strategies In Product Design, Manufacturing, and Management, eds. K. Wang, Kovacs G., Wozny M., Fang M., (Boston: Springer), pp. 597-603.

This paper focuses on how to build an effective and preferable embedded system network and select some suitable tools to finish their different works collaboratively. And, here we give a general project to solve the CSCW problem of embedded systems in final.

2. EMBEDDED SYSTEM AND CSCW

Embedded system [2] is a special type of computer system that is centered on application, adjustable in software and hardware and suitable to the strict application requirements in function, reliability, cost, size and energy consumption. The embedded computer well surpasses the general computer in the extent of application. They also are widely used in manufacturing industry, process control, communications, instruments and meters, automobiles, ships, aeronautics and astronautics, military equipment, and consumer goods. The strong points of embedded system are of good real-time performance, low-cost, miniaturized, specialized, high reliability, extensible and configurable.

Talking to CSCW, we must say something about the birth of CSCW-Office automation. In the mid 1970's, single user applications such as spreadsheets and word processors ran amok on mini-computers. Office automation was attempt to integrate those applications, but without any real idea of the user requirements or of the sociology of sharing the source documents. CSCW then began as an attempt by software engineers to incorporate into new software tools know on how people work in groups. This knowledge was obtained form practitioners in the fields of, for example, social psychology and organizational theory [3].

3. COMMUNICATION OF EMBEDDED SYSTEMS

3.1 Connection to the Intranet/Internet

Considering the performance and the cost, you must estimate the resource demands earlier. It includes the memory demands of protocol stacks. Using more protocols will make you system bigger. And you must consider what hardware is needed in your system, maybe an Ethernet controller, or a UART asynchronous adopter and a serial interface, or even a wireless receiver. These factors should be considered in your project [4].

And you must consider the additional consumption to your CPU added by the protocol stacks, because adding a network support are likely to make your system sluggish. If you bought the protocol stacks from a RTOS

developer, the stacks are integrated in the operation system. The developer will afford the technical support.

There are many models that connect embedded systems to the Intranet/Internet for different network demands and construction purposes. But generally speaking, we may just consider how to implement the TCP/IP network protocol. The two questions of connecting embedded systems in the Intranet/Internet are the media selection of information transmission and the protocol selection. Oriented to the two questions, the most common connecting method is: using the Ethernet and TCP/IP protocol, achieve physical layer and link layer protocol through network interface controller (NIC), and at the same time the micro-processor run the TCP/IP protocol communication module to complete the connection with Intranet/Internet.

3.2 Preferable bus architecture

Many communication styles can be used in the embedded systems based on micro-processor. It is different with the speeds and times in communication processes. The most common transmission is the communication between micro-processor and memory, and that the low-speed communications between micro-processor and peripheral equipments are less. All communications can be done in one high-speed bus. But it has the following disadvantages: First, every peripheral device must have high-speed bus interface, which will add logic switches, power waste and cost; Second, generally high-speed bus are particularly designed for special processor, so the peripheral devices with the interface above are not easy to be transplanted; Third, the speed may be greatly reduced when the peripheral devices in the bus are too many.

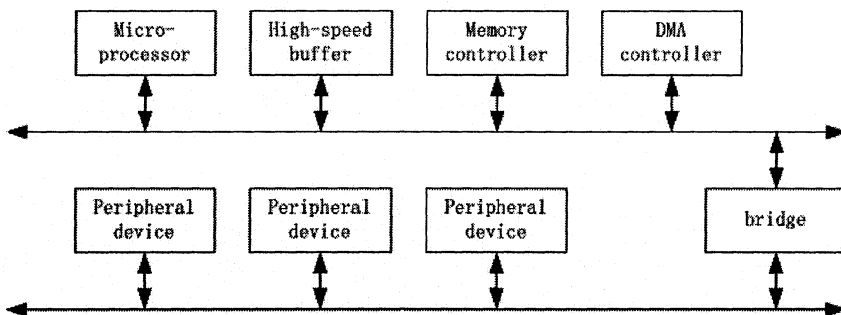


Figure 1. Two layers bus architecture.

For the reasons above, the network can be designed to be with two layers (Fig.1). It is the high-speed local bus and the low-speed peripheral bus. The high speed local bus is used to connect micro-processor, high-speed buffer,

and memory controller and high-speed co-processor. It is a special bus and it is generally as wide as the width of a word of the memory.

The peripheral bus should be designed to have a good transplanting performance. It should have a low power-waste, and have less logic switches. It is generally compatible with the standard industry bus, such as ISA and PCI, so it can be transplanted easily. It is narrow and low-speed compared with the local bus, so it doesn't need many logical switches, and doesn't consume much energy. And now there is some other peripheral bus currently used, such as CAN, SPI, which has a higher speed [5].

The bridge is a single-purpose processor; which can transfer the communication with this bus standard to the communication with another bus standard.

Sometimes, the architecture can be built with three layers. The first layer is the micro-processor local bus, the second is system bus, and the third is the peripheral device bus. The system bus is high-speed bus. It undertakes some communication tasks in the local bus. This three-layer architecture is beneficial to build some complex systems with multi-coprocessors.

3.3 Wireless communication

Besides the traditionally serial communication and parallel communication, wireless communication can avoid the physical contact with the networked devices. The physical layer used in short-range wireless communication is IR (infrared ray) channel or RF (radio frequency) channel. There are three famous wireless protocol: IrDA (Infrared Data Association), Bluetooth, and IEEE802.11. They are widely used in embedded communication devices. In a long-distance communication, GSM/GPRS is the selection.

4. DISCUSSION OF CSCW METHODS FOR EMBEDDED SYSTEMS

4.1 CSCW tools for embedded systems

To date, the technology used in CSCW can be broken down into several areas. By examining the functionality of the available tools in each area, it is apparent that the distinction between each is blurring towards a few hypothetical, homogeneous "super tools". The "super tools" would cover E-mail/Messaging, Groupware, Conferencing, Software Development and Virtual Rooms. The common theme in what is available is the underlying use of the Internet, and in particular the Web, as an inter-personal computer connection transport.

Table 1 shows the diversity of CSCW categories as they appear today. Comparisons are made between embedded systems and general computers. For each category there is an estimate of how widely it can be used by the embedded system, and some example tool names given.

Table 1. CSCW categories can be used in embedded systems

Category	Can be used in embedded systems	Example projects and tools in PC
Bulletin Boards	Yes, Easy	WebLines
Chat	Yes, Easy	WebChat, NetMeeting
Collaborative Design	Yes, Difficult	n-dim
Collaborative Drawing and Writing	Yes, Difficult	GroupSketch, Conversation Board, Emacs with
Collaborative Software Management	Yes, Difficult	CVS, RCS, SCCS
Collaborative Learning	Yes, Difficult	MUNIN
Email	Yes, ubiquitous	Numerous
Groupware toolkits	Yes, Difficult	Lotus Notes, Microsoft Exchange,
MUDs	Yes, Not difficult	XXM, XTV
Newsgroups	Yes, Easy	Navigator, Internet Explorer
Virtual Labs and Rooms	Unkown, Very difficult	TeamWave Workplace, UARC, AAEM
Share Windows	Yes, Difficult	NCSA Xcollage(defunct)
Videoconferencing	Yes, Difficult	Vic, vat, rat, nv, CU, NetMeeting, WebPhone, VidCall, FreeVue, Vfone, etc.
Virtual Reality	Yes, Very difficult	VENUS, CAVE, DIVE
Web-based conferencing	Yes, Difficult	WebNotes, NetMeeting
Web-based calendars	Yes, Easy	WebCal

Pure videoconferencing is technology that allows two or more users to interact using audio, video or both. It is covered by a set of standards relating to CODEC, ISDN and packet-based conferencing, which are more or less supported by the available tools. And Virtual labs and rooms contains tools that allow users to enter virtual spaces, manipulate objects within them, and interact with other users “present” in the space. Meeting ware is for tools which integrate videoconferencing as part of a suite of subtools to support meetings. Because these tools need high bandwidths and abundant resource, it is very difficult to be transplanted to embedded systems.

Workflow tools or groupware help in the management of project schedules. They allow the sending, reception and annotation of electronic documents, and the triggering of messages to be sent when certain conditions have been met. They can be easily to be transplanted to embedded

systems for their flow-style mechanism. But they should be cut down to meet the embedded systems.

Email and messaging is the most widely used and platform-independent groupware system. It is the lowest common collaborative computer tool. Embedded system can make heavy use of Email, and it is not further treated here, and some embedded systems have included it. Usenet news can also be widely used in embedded system. Most embedded Web browsers have built in News readers. Bulletin board systems were mainly designed in the late '70s as dial-up systems dedicated to the exchange of files between people. There are Web-based BBS systems, such as WebLines, available. In general, BBS systems can also be used in embedded systems. Multi-user dungeons (MUDs) are also referred to as Multi-User Dimensions. The basic idea is that, using a text-based system, the user enters an environment that contains objects and other users, and then interacts with the environment by typing commands. The main use of MUDs is in game playing coupled with social intercourse between the participants. And it might be a synchronous tool can be used in embedded systems [6].

4.2 Synchronous or Asynchronous

Do CSCW tasks intrinsically need high bandwidths? To answer this question, divide CSCW applications into synchronous and asynchronous. A synchronous CSCW requires communications facilities as the system is being used; an asynchronous CSCW requires communications facilities between users. Thus synchronous CSCW permits users to work on the same task simultaneously; whereas tasks in an asynchronous CSCW must be interleaved. There are clearly advantages in both approaches, depending on the exact nature of the tasks being supported. It is clear, however, that the communications media for the alternatives need not be the same, in particular, synchronous and asynchronous CSCW have very different maximum and average bandwidth requirements considered as a function of the number of active participants in any task. For embedded systems, considering the limited bandwidth, MUDs might be a good synchronous tool, and Email, Bulletin board systems might be good asynchronous tools.

5. FUTURES FOR CSCW IN EMBEDDED SYSTEMS

What is driving the market for the paper talked above? As embedded systems become more and more networked, small groups of workers become potential new customers for groupware tools. Mature single user embedded applications can be embellished with groupware features. Telecom companies seek to increase demand for networks by promoting multimedia tools that make use of them. Commercially, the platform being targeted to

the embedded systems are rapidly developed, and the corresponding CSCW tools are unavoidable to be put on the table.

Apart from commercial factors, there is considerable interest in the academic community in wide area project collaboration, and the need for tools that support collaborative work at a distance is evident when considering the size of new generation industrial embedded systems collaborations and their geographical spread, such as meeting-ware and shared applications. Transplanting existing CSCW toolkits to embedded systems to support wider area collaboration looks promising for the future [7].

6. CONCLUSIONS

This paper discusses CSCW methods that can facilitate the work of embedded systems. It focuses on how to design and build embedded network architecture, and which CSCW tools should be selected in embedded systems. Generally, TCP/IP protocol and two-layer bus architecture are preferable. The CSCW tools such as MUDs, Email, Bulletin Boards and so on can be used in embedded system. It is evident that the CSCW for embedded systems has a promising future.

7. REFERENCES

1. John V Farr, Dennis M Buede, "Systems engineering and engineering management: Keys to the efficient development of products and services.", *Engineering Management Journal*, Vol 15, No. 3, pp. 3, Sep. 2003
2. Rick Grehank, Rober Moote, Ingo Cyliax, *Real-Time Programming*, Addison-Wesley, USA, 2001.
3. Junzhong Gu, *An Introduction to Computer Supported Cooperative Work*, Tsinghua university publisher, Beijing China, 2002.
4. Frank Vahid & Tony Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, USA, 2002.
5. Hassan Gomaa, *Software Design Methods for Concurrent and Real-Time System*, Addison-Wesley, USA, 2003
6. Yulong Wang, "Open architecture for CNC controller and their developing prospect", *CNC & Software*, pp. 75-76, Apr 2004
7. Shuzi Yang, "Trend in the development of advanced manufacturing technology", *Chinese Journal of Mechanical Engineering*, Vol 5, No. 1, pp. 73-77, Dec. 2003.