Production Systems and Information Engineering
Volume 2 (2004), pp. 131-142

HIAC
HIERARCHICAL INTER-AGENT COMMUNICATION SYSTEM

FERENC VAJDA
Mobile and Microrobotics Laboratory
Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
vajda@iit.bme.hu

[Received November 2004 and accepted January 2005]

Abstract. In our century, there seems to be more and more demand of multi-agent heterogeneous robotic systems whose agents together can fulfill tasks that had been very difficult to apply and required deep programming knowledge earlier. The most difficult task about heterogeneous systems is perhaps to implement the right communication between the devices. Although IP-based communication can hugely help to solve this problem, it is not really applicable in many respects for a robotic system whose structural, hierarchical architecture makes the construction of a special communication system expedient. The main aim of our article is to give a recommendation to the communication of the systems that have a structured device set of huge number of entities being in close relation to each other.

Keywords: Communication, Protocol, Hierarchy, Control, Multi-agent Robotics

1. INTRODUCTION

HIAC [1] is a communication stack of on-line systems. So first of all HIAC provides tools with a relative small amount of information for higher level communication. However, transmitting larger amounts are also possible, but this was of smaller moment in the course of development. HIAC supports the modern communication set of processes (or independent agents in HIAC) on higher levels. The easy implementability was one of the most important standpoints, so it is possible to implement HIAC on devices to which implementing more complicated protocols would be very difficult or impossible. This ability must not lead to smaller number of features if communication is led by devices with larger computation capability. It is important to note that the HIAC is suitable for handling the communication of systems that fulfill a common task or tasks, and not for making a huge number of operations that are totally independent of each other – as IP (of Internet) [6] does.

This article is a short outline of the HIAC protocol stack, and it tries to introduce the main components of this communication system. We tried to point out some significant aspects in a more detailed way to show the importance of the development of the stack.

Testing the communication system is now under process in the frame of the Balaro [3] project. The hierarchy of the system is given by the architecture of the system.

The HIAC is still under development, so some questions will be able to get answered by the time the stack has been completely developed. Such questions are the enhanced gateway functionality of the participant devices, the more secure data transfer, or the inter-agent communication of mobile robotic platforms, etc.

1.1. Relation to the ISO-OSI model

HIAC covers multiple layers in the ISO-OSI model [7]. First of all HIAC is a set of protocols built over each other, moreover the higher level protocol covers the tasks of multiple layers. The two protocols are EDCP (Escape Driven Connection Protocol) and HIACP (Hierarchical Inter-Agent Communication Protocol) which does the substantive work. HIACP is built over EDCP, which supposes a two-way on-line connection between the devices. The connection may be indirect, so EDCP can operate over an open TCP port, RS232 channel, or any continuous open connected channel. The next figure shows the location of HIAC in the ISO-OSI model.

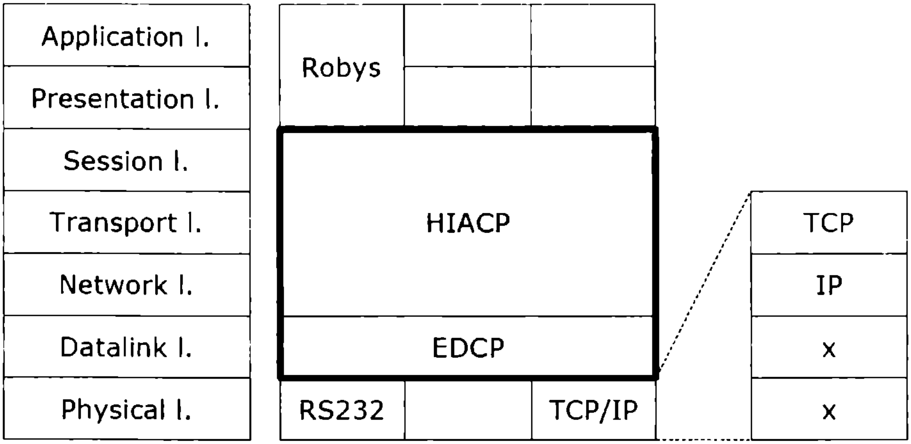

Figure 1: Location of HIAC in the ISO-OSI model

There are no restrictions for higher levels, but it is important to note that HIAC is in a close connection with Robys [2] – hierarchical, parallel, object- and taskoriented [4] robot-programming – system, which is under development. Robys will be suitable for heterogeneous robot systems or even for distributed parallel systems.

2. EDCP

EDCP is a rather simple protocol, which helps the HIACP to separate packets and to monitor the continuous connection, etc. EDCP represents the datalink layer of the HIAC Protocol stack, so it realizes the point-to-point connection between two "direct" connected devices. This datalink layer is simpler than that of other systems (it contains not even a checksum), because it assumes quite a safe data-transfer (the layer beyond and the higher levels of HIAC take care of it).

The octet (or byte) based protocol is escape-driven. This means that some preset – so called escape – characters (EDCP codes) have special objectives while the other octets get through the protocol transparently. If we would like to transfer octets that have special meanings in the protocol, they will be transfered by the help of a distinguished EDCP code. Some EDCP codes require further information that are located after the code as additional octets.

2.1. EDCP codes

The 00-F7 range of octets makes the movement of the characters of the corresponding value that should be transfered in a higher level, and the F8-FF range is reserved for EDCP codes. SPKG (0xF8) starts and EPKG (0xF9) closes each packet. This makes the data transfer robust, because this way we know the start and the end of the packet, there cannot happen any misinterpretation or overrun, etc. as the consequence of lost data. The checking whether all the data of the packet has arrived without failure is done by the higher layers. If we want to transfer F8-FF code corresponding data in a higher level we have to come up with a special solution. In these cases the data transfer is helped by the EXCH (0xFC) code. The least significant 3 bits of the octet after the code determines which F8-FF character will be transfered. The value of the higher 5 bits: 11001. If the octet after EXCH is different from this, EXCH will be ignored.

2.2. Connection Monitoring

The EDCP does not give recommendation on building up or cutting off the connection; however, since it assumes continuous connection, this necessitates certain complementation. Since it may occur that a device is not sending data for a longer time, the communication partner does not know if the connection has broken or not. The EDCP specifies the maximal period of time within which the partner has to send any data. If there is no information to give they can also send an empty packet (SPKG+EPKG).

2.3. Overhead

Similarly to other protocols the EDCP has a certain amount of overhead. For huge packets with data that have even statistical distribution (e.g. longer messages) its value is around 4-5%, while for small packets (e.g. signals) it may be 20-25%, and in certain cases even much bigger then these. In practice, by choosing the suitable communication parameters of the higher level HIACP, the overhead can be held around 20% for the shortest packets as well. Using F8-FF coded characters is also avoidable for longer messages in most cases.

3. HIACP

The HIACP is a much more compound protocol. Besides the fact that it supports complex operations regarding a structural hierarchy, it provides minimal implementation in simple devices. We can send signals and messages, we can use shared memories, semaphores or synchronization through the protocol independently of hierarchical levels.

3.1. Structure

HIACP is a packet based protocol. This means, that each device of the system provides their information, commands, etc. in a packet to the partner device. We can use various types of packets for the different tasks. These packets are similar to each other in several aspects, but their content carries different aims.

Generally, during communication the devices have a direct contact with several other devices in a hierarchical arrangement. The possible target and the content of the messages depend on the level and the location in hierarchy.

Since it is not sure that the two devices that want to communicate with each other have direct contact, certain devices must have a gateway function, as well. Some devices can interpret the protocol in a higher while others in a lower level. Therefore, it may happen that the devices without a direct contact will try to communicate with each other in different way. (The devices in direct contact are equal in the set that both of them supports, thus their communication is not problematic). The task of the gateway is not only to transfer message packets, but also to change the packet itself based on the suitable communicational abilities. These changeable parameters include the type of checksum, response to the message, etc.

Each device has its own unique identifier, which determines its position and role in hierarchy. This identifier is the HIAC address. This is completed by an identifier (HIAC mask) by which we can specify with whom it can communicate, which is its workgroup, etc.

Despite other communication protocols we can surely assume that the devices have a lot of a priori information. In the case of other systems the abilities of each system element appears only in a higher level, while in the HIAC we should know the roles and abilities even in a lower level. In the initialization phase the devices share their important communication parameters, but there are several data which contain no relevant information for the communication partner. These information does not change too often in systems where using HIAC is expedient.

3.2. Hierarchy

One of the biggest advantages of the HIAC is the hierarchical structure. Thus, each device can communicate only with partners meeting its hierarchical level and position, and this way we can more easily supervise and handle the controlling of a system.

Based on hierarchical levels we can distinguish three communication directions: the *employer*, the *employee* and the *colleague* communication. Although the employer and the employee communication has a close relation, we have to make difference between them, since an employer can say different things to its employee, than vice versa.

3.2.1. Employer Communication

Under the name of Employer Communication we mean the communication with the higher ranked device that is directly above the other device. Although, in the course of employer communication certain regulations have to be observed when talking to a higher ranked device, most of the communication methods are open. The communication has to be initialized by the employer in any case (INCON – initialize connection). It cannot be addressed, until it initializes the connection. It is also not possible to give certain synchronization commands. Although, it is not regulated in the specification, when making the system plan we should not implement commands that are given by an employee to its employer.

The device cannot talk to its employer's employers. It can solve this situation only by „asking" its employer to „settle" the thing with its own employer.

3.2.2. *Colleague Communication*

Under the name of Colleague Communication we mean the communication with devices of the same rank, in the same workgroup. Here we have more than one possibilities: any of the two parties can initialize the connection, and use the synchronization. The HIAC specification does not give smaller rights in the course of colleague communication than employer communication. However, it is important to take care when designing the higher level system that the colleagues cannot give commands to each other only information. There may be certain warnings that can be sent to each other.

3.2.3. *Employee Communication*

Under the name of Employee Communication we mean the communication with the lower ranked device that is directly under the other device. This communication direction includes the widest possibilities set of communication types.

In the course of this we can use any forms of communication, including that we can insert the lower ranked device into the network, remove it from there or we can pass it over another device.

3.2.4. *Subemployee Communication*

Subemployee communication is a special instance of employee communication. Under the name of Subemployee Communication we mean the communication with the lower ranked device that is *not* directly under the other device. At present this is a one-way communication: the device may give commands to which it does not expect a response, to the subemployee. If it wants to directly control the device it can take away the subemployee from his own employee, but if so, the subemployee's original employer cannot give commands to it.

Subemployee communication can be important especially in systems using mediation hierarchy [5], which means that users may intervene on all the hierarchical levels (from high-level control down to the low-level physical layers) of a robotic system.

3.3. Packet Structure

The HIACP is built up of packets. The structure of the packets is the following:

	7	*6*	*5*	*4*	*3*	*2*	*1*	*0*
1	PV				PT			
2	-	PRI			CR	MP	CT	FA
3		PSN (*byte* / none)						
4	TRG (*dword* / *adaptive*)							
5	SRC (*dword* / *adaptive*)							
6	LNG (*adaptive*)							
7	DATA (...)							
8	CHK (*byte* / *dword*)							

Figure 2: HIAC packet architecture

PV (Packet Version) – version number, at present always '1'
PT (Packet Type) – type of packet, see more in 3.5
PRI (Priority) – priority of packet, lower values mean higher priority
CR (Confirmation Required) – confirmation is required if value is 1
MP (Multiple Packet) – multiple packet mode, numbering packets, fragmentation
CT (Check Type) – type of checking, simple sum or CRC32
FA (Full Addressing) – full addressing is used, see more in 3.4
PSN (Package Serial Number) – automatically increasing serial number of packets
TRG (Target) – target's address
SRC (Source) – source's address
LNG (Length) – number of data octets
DATA (Data) – any information, depending on packet type
CHK (Checksum) – checksum

3.4. Addressing

Every device of the HIAC system has a unique identifier, the HIAC address. The HIAC address is a 32 bit identifier, whose value depends on its position and role in the system. The HIAC system has a hierarchical structure, which means that each device's role in the system determines with whom and in what level it can maintain connection. Since this connection belongs to a subnetwork part, it is enough to specify one subnetwork identifier, level address in the course of the addressing process. This identifier is part of the HIAC address, in other words the level address is the value which is masked by a so called HIAC Mask. We have to define the HIAC address and the HIAC mask by dividing these values into octets. These octets are represented by two hexadecimal digits, and they are separated by ':' characters from each other.

E.g. If the HIAC address of a device is 3C:12:F8:00 and the value of the HIAC mask is 00:0F:FF:00, the level address is 2F8.

Every device gets its HIAC address and its HIAC mask from its employer. Thus, it will know with which devices it can communicate. In a certain subnetwork, the devices above certain other devices have always 0 level address and every other value implies the addresses of devices of the same rank. Besides the 0 level address of the subnetwork, its highest valued level address also has a special role. This is a so called broadcast message, which has to be interpreted by every participant as if it had been meant directly to them.

The address distribution also follows the hierarchical structure of the system (see fig. 3). This means that an employee may only have such an address which is derived from the employer, i.e. the employees constitute one ore more subnets of the employer. For example, if an employer's address is 20:C4:D2:00 (e.g. with mask 00:00:3F:00) employees' addresses can be 20:C4:D2:01-20:C4:D2:FE (e.g. with mask 00:00:00:FF). 20:C4:D2:FF is the broadcast address of the subnet.

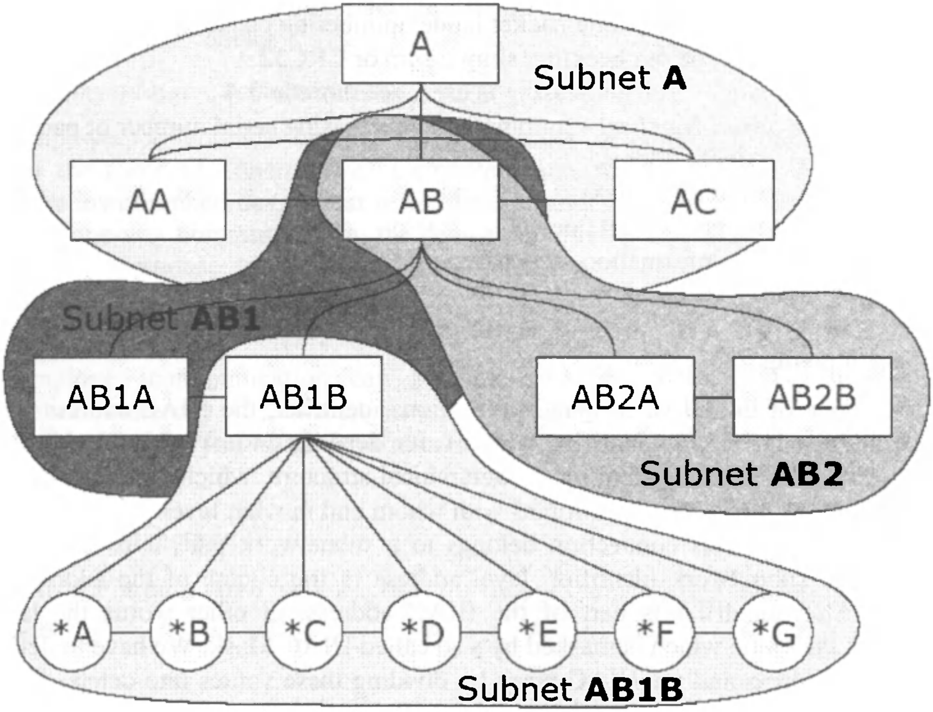

Figure 3: Hierarchical architecture of HIAC addressing

In the current version of HIAC employers are “directly” connected with their employees (it may also be a TCP/IP connection). Colleagues are normally not connected, so if two colleagues want to communicate with each other, they send their messages through the employer. Note that this simple gateway functionality of the employer is a “low level” task, so the employer does not care about the message.

3.5. Packet Types

The content of the data field depends on the packet type in any case. Within this article we cannot give a detailed description of each type and the information transfered by them. However the following table briefly summarizes the available types.

Table 1: Packet types

TYPE	***Name***		***Description***
1	SIG	signal	sending quick warnings
2	SMPH	semaphore	testing for free resources
3	MSG	message	sending commands, appeals, information
4	SHMW	writing shared memory	storing general/common information
5	SHMR	reading shared memory	
8	SYNC	synchronization	synchronization, parallelization
14	SYS	system packet	system specific data transfer

3.5.1. System packets

System packets fulfill the fairly important tasks which are necessary for operation of HIAC, e.g. passing over data transfer parameters, distributing hierarchical arrangement, gateway functions, etc. We have to specify a subtype within the type of the system packet – this subtype will determine the exact task. The following subtypes are the most important ones:

Package Confirmation (PKGCNF): Response to a packet. May refuse, indicate failures or may accept.

Initialize Connection (INCON): The communication initializing partner passes over the communication parameters supported by it, and the HIAC data (address, mask) of the target device.

Accept Connection (ACCON): The communication receiver partner passes over the communication parameters supported by it, indicating that it is ready to communicate.

Refuse Connection (REFCON): If for some reason the device is not ready for communication (e.g. The employer wants to establish an employer-employee connection, but the employee has already an employer)

Reinitialize Connection (RECON): Necessary if the position of the device in the hierarchical system changes; therefore, its HIAC data and even the abilities of the communication partner change.

Employee Transfer (EMPTRF): If an employer does not need its employee it may pass it over to another device. Passing over is indicated by EMPTRF.

Workgroup Transfer (WKGTRF): Similar to EMPTRF, but a whole workgroup can be passed over.

Employee Request (EMPREQ): Request for an employee. An employer may instruct an employee to pass over one of its own employees. This can help if a device starts to have defective behavior. Its workgroup(s) may work onwards. (If there is a greater fault – and the device is out of order –, directly calling the subemployee is also possible, but this is not the question of EMPREQ.)

Workgroup Request (WKGREQ): Similar to EMPTRF, but a whole workgroup can be asked for.

4. APPLICATIONS

HIAC was developed mainly for robotics related applications, i.e. multi-agent heterogeneous robotic systems; however, it is also possible to use the protocol stack in other areas using hierarchical architecture of coupled devices, applications, etc.

The first system on which this communication system is tested is the Balaro [3] project. The Balaro (Balaton Rover) is being developed for mapping the basin of Hungary's largest lake, the Balaton, in order to find and pull out dangerous objects, relics from the second world war, etc. There is a simple hierarchy in this system, but it contains all the possible communication types, so it is ideal for testing and using HIAC. The Balaro system comprises a mobile platform (later there will be more of them) wandering on the ice cover of the lake and a station on the coast. The highest level of this hierarchy is the coast-station, all the lower levels are built in the mobile platform itself. The mobile platform has a control unit, which is responsible for the proper operation of Balaro, and some lower level intelligent devices: the sensor system, the GPS with track-recording, the driving, etc. The devices are connected through TCP/IP connection (wireless with the coast station). The coast-station is the employer of the control unit, which is the employer of the lower level devices, so the devices are the subemployees of the coast station. The devices are colleagues of each other. The employer, employee and subemployee communication need no further explanation. Communication between colleagues is also necessary, e.g. the sensor system can "control" the driving directly in case of danger.

Other applications are in view in the field of microrobotics and other multiagent mobile systems, and also in the field of robotics on the game-market.

5. FUTURE PROSPECTS

Working out the specification we did not pursue to create an all-comprehensive system. There are several further development possibilities that can be integrated in later versions of HIAC. Some of the development prospects are the followings:

- *Secure transfer*
  A data transfer is secure if unauthorized devices cannot access data and even they cannot send data in the name of a partner in the system. SSL or other public-key encryption systems can be taken into account. If secure data transfer is needed now EDCP can also work over an encrypted connection.
- *Gateway function*
  Basic gateway functionality is already given in HIAC, but it needs several further supplements. If a device receives a message and it knows where to forward it (i.e. there is a direct connection with the target), it must forward it. If it does not know the target device, it has to forward it to its direct employee. This way of gateway functionality is mostly enough in communication with direct connection. In some cases this is not so. If using a system where the system's devices are not in a hierarchical arrangement (e.g. token ring), or a mobile system, where devices are moving continuously, the gateway theory must be expanded.
- *Moving systems*
  In mobile robot systems, often the devices cannot communicate on-line with each other, only when they meet somehow. Answering these questions remains to the later versions of HIAC

ACKNOWLEDGMENTS

The author gratefully acknowledges the contributions of the Hungarian Scientific Research Fund (OTKA T 042634).

REFERENCES

[1] VAJDA, F.: *HIAC$_{VI}$ Specifikáció (HIAC$_{VI}$ Specification)*, MoMic, dept. of CEIT, BUTE, 2004

[2] VAJDA, F.: *Robys – a Hierarchical Robot Programming System (Developed Mainly for Multiagent Microrobotic Environments)*, CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby, Vienna, 2004

[3] VOGEL, M: *Balaro: Semi-autonomous Ice Rover For Localization of Unknown Objects in Frozen Lakes*, CLAWAR/EURON Workshop on Robots in Entertainment, Leisure and Hobby, Vienna, 2004

[4] VAJDA, F., URBANCSEK, T.: *High-Level Object-Oriented Program Language for Mobile Microrobot Control*, Proc. of the INES 2003 International IEEE Conference on Intelligent Engineering Systems, Assiut-Luxor, Egyiptom, 2003

[5] ADAMS, J. A.: *Human management of a hierarchical system for the control of multiple mobile robots*, Dissertation, University of Pennsylvania, 1995

[6] POSTEL, J.: *Internet Protocol – DARPA Internet Program Protocol Specification*, RFC 791, USC/Information Sciences Institute, 1981

[7] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *ISO/IEC 7498-1:1994 Information technology -- Open Systems Interconnection -- Basic Reference Model: The Basic Model*, Ed. 2, 1994