APPLICATION OF REACTIVE AGENTS IN AUTOMATIC BINDINGS OF LONWORKS NETWORKS DEVICES

Miguel dos Santos Alves Filho, Rafael de Aquino Cunha, Carlos Eduardo Cugnasca
Escola Politécnica, University of São Paulo, Av. Prof. Luciano Gualberto, travessa 3 n° 158, São Paulo, SP, Brasil
miguel.alves@poli.usp.br, rafael.aquino.cunha@poli.usp.br, carlos.cugnasca@poli.usp.br

Keywords: LonWorks®, multi-agents, Artificial Intelligence, distributed systems, PBX, telephony, control networks.

Abstract: This paper has the goal to propose and test a new method to implement dynamic bindings in LonWorks® technology, allowing a new Distributed System of Private Telephone Switching (DSPTS), also developed with LonWorks® technology, to make their telephone links. In order to do this, a method for developing embedded systems and the reactive agent view was applied for each different device in this new system, offering a unique, practical and innovative solution for both, LonWorks® and PBX systems. This view allowed the implementation of intelligent and autonomous devices, specially in their internal process, granting satisfactory and more efficient results based on the DSPTS requirements. This work is the kick-off and the basis for developing new functions for telephone systems and control networks.
1 INTRODUCTION

In recent years, the paradigm of control systems design has been changed, moving from the traditional centralized architecture and proprietary technology to distributed and open architectures (HUR; KIM; PARK, 2005)(PU; MOOR, 1998).

This change of paradigm takes the concept of network control, and brings some benefits to automation systems, such as reducing and simplifying the cabling, increasing reliability and more options of manufacturers and integrators for the user to choose (ECHELON CORPORATION, 2007) (HUR; KIM; PARK, 2005).

Aiming to exploit these advantages this paper presents a research project that proposes to develop a Private Automatic Branch Exchange (PABX) (now also called Distributed System of Private Telephone Switching - DSPTS), with distributed architecture and implemented with the control networking technology LonWorks®. In this architecture all telephone extensions and devices for interfacing with the telephone line are implemented as devices (also called nodes) of control network that communicate with each other, with the audio signals (voice) digitized and transferred via messages.

The standard method of use of such technology and the available tools to work with it provide only a static way for the realization of logical connections between devices (called bindings). It means that these connections are defined during the setup project and then will remain unchanged (ECHELON CORPORATION, 1999).

Considering that a PABX system must open and close communication channels every time, it means that the logical connections between nodes are not pre-defined, but defined as the need to establish a telephone connection. So the use of this control networks technology could not be done in its original form.

The aim of this work is to propose and analyze a solution based on the concepts of multi-agent systems to develop devices of this distributed PABX, using the LonWorks® technology, with self-management in a previously configured environment, enabling the creation of dynamic telephone links for the DSPTS.

The DSPTS characteristics of a distributed application stimulated the use of the multi-agent system concepts in its implementation (DURFEE; ROSENCHEIN, 1994).

1.1 LonWorks® Technology

LonWorks® emerged in the 90s, developed by Echelon. This is a technology for automation networks, involving not only a communication protocol, but also all the necessary infrastructure for the system development as management tools, products development tools and network configuration tools (ECHELON CORPORATION, 1999).

Its protocol was developed based on the OSI reference model, from the link layer to the session layer. It is an open and standardized protocol, ANSI / EIA 709.1, also known as LonTalk (ECHELON CORPORATION, 2002).

An emphasis point of this protocol is its messages structure, which allows direct messaging between devices, broadcast messages and multicast messages. Furthermore, it provides system management messages, allowing a greater control of devices and even loads and develops programs by the network itself (ECHELON CORPORATION, 2002).

In the implementation of control networks can be used various media types, as well twisted pair, power network, optic fiber and wireless communication, which makes the technology very flexible (ECHELON CORPORATION, 1999) (CHERMONT, 2007).

The LonWorks® control network nodes may be treated as objects that communicate through interfaces, also called functional profiles, which has network variables of the device in question. It is through these variables that information may be shared with other devices (XIE; PU; MOORE, 1998).
The communication between the nodes is done through links between their network variables, process called binding. From them, they use logical connections that allow messages to be sent whenever there is change in the network variables values (ECHÉLON CORPORATION, 1999).

Besides the LonWorks® technology design, Echelon has developed a chip called NeuronChip that is a full implementation of the LonTalk and provides the basis for any manufacturer to develop its products.

The NeuronChip consists of three processors working in parallel, two of them for the treatment of the communication protocol and the last one for the application program execution (MOTOROLA, 1997).

The use of NeuronChip allows developers to treat the network variables as program variables, aside from the whole network communication, developing the programs associated with the application (MOTOROLA, 1997). It was developed a specific programming language to that technology, called NeuronC, which follows the ANSI C and is completely based on events (ECHÉLON CORPORATION, 1995a).

### 1.2 Reactive Agents

Agents are autonomous entities that work together to achieve the same goal, being able to interact and to organize efficiently (LIVIU; SEAN, 2004).

They also can be perceived as real or virtual automatons that have knowledge of the environment in which they are inserted, and are able to perceive changes in that environment. They have knowledge about other agents, being able to communicate, to learn, to infer, to form groups or even to reject. Finally, they are able to decide, as his observations, goals, knowledge and interactions (WOODRIDGE; JENNINGS, 2002).

The most common way to classify these agents divides them into two groups, considering the deliberation capacity, the environment perception and communication complexity: one is called cognitive agent and the other, reactive agent.

The cognitive agents are characterized by having very complex functions and by having models that require high processing capacity (WOODRIDGE; JENNINGS, 2002).

There are different internal architectures that can be used for the development of such agents, being one of the most used architecture known as “Beliefs, Desires, Intentions” (BDI) (RAO, 1996) (RAO; GEORGEFF, 1991). Its use is considered in systems that require the ability to exchange complex information, to own and build complete models of the world where they belong, their own and other agents, and thus acting spontaneously, creating an organization that serves a common purpose to all of them (STEELS, 1990).

A good example of the use of cognitive agents is presented by Bigham (2003), which consists of an antennas chain for mobile phones, each one with a manageable coverage area, so that intersections between them can be done and undone.

Within the group of reactive agents there are those who are able to understand (though fairly limited way) and to react by acting on the environment in which they are inserted, through a pre-defined logic and always with a final goal that was set in project phase.

A striking feature of this model type is its simple communication way, which often occurs in indirectly, through the environment itself (WOODRIDGE; JENNINGS, 2002). One advantage of these agents is the ease of implementation, which may be based on devices with less processing capacity and great limitation of energy.

As examples of its use may be mentioned the study of the behavior of insects and their development processes as shown in Liviu e Sean (2004), or control networks devices with low processing capacity. Finally, this class of agent is used in systems where the intelligence is expected to arise from the society overall behavior and not from each individual (STEELS, 1990) (CASTELFRANCHI, 1998) (BOISSIER;
It is worth mention that the software agents are used in systems known as multi-agents systems or distributed problem resolutions, which have as one of its characteristics the lack of centralized control, which falls within one of the DSPTS requirements as well its simplicity, allowing its implementation in the NeuronChips (DURFEE; ROSENCEIN, 1994).

1.3 The DSPTS

One of the most important systems for any enterprise is the voice communication system. Currently, these systems tend to be implemented with equipment called PABX, which are nothing more than core private telephony, which enables internal communication between all employees who have access to branches, and communication between them and the outside world (external links).

One important feature of this equipment is its centralized architecture, in which all telephone extensions and external lines must be connected to a central device, as illustrated in Figure 1. Among the main limitations of this architecture there is the limit of the expansion in the extensions number and external lines, and the need for a large quantity of cabling, which makes it not very flexible to install and to change the branches positions.

The research presented in this article proposes a new architecture for the implementation of such systems, using the technology for control networks LonWorks®, allowing to remove the cited disadvantages of typical PABXs, as well the reduction and simplification of cabling, flexibility for changes, incremental growth (and associated investments), and greater system reliability (gaps in some nodes do not prevent the functioning of the rest of the system).

Moreover, this project aims to add greater intelligence to these systems, solving problems such as call direction, to decide which carrier should be used according to the call type, extensions calls prioritization to access the outside line, etc.

Figure 2 illustrates the architecture of the proposed system, in which are present the following DSPTS components:
- TLM – Trunk Line Module;
- E1M – E1 Module;
- ARM – Audible Response Module;
- DTM – Digital Telephony Module.

Other features that motivated the choice of the LonWorks® technology was the possibility of the object oriented programming and the low jitter in messages exchange, due to the robust features of the communication protocol.

To the voice messages exchange, the project was specified for the use of voice compression algorithms, aimed at telephony. In this case, was used the G.729 ADPCM algorithm, which provides a good quality for telephony voice communication and uses a low bandwidth (16 kbps per communication channel).

Just as occurs in conventional PBX, it is necessary for the DSPTS to close communication channels, with the difference that in the first case the link is physical and in the second, it is logical.

In the LonWorks® technology this logical link is performed by means of bindings between the
variables of two devices that as reviewed above, are built with the tools on the market, during the system configuration. This makes these static links, preventing the intended application.

Thus, the possibility to perform bindings dynamically is fundamental to the DSPTS because that way two devices could exchange messages during a phone call, after which they would be free to communicate with other devices and participate in other connections.

2 MATERIALS AND METHODS

The nodes developed have low processing capacity and low memory amount, and because it is a research project, in which new requirements can always be discovered during the evaluation of prototypes, it has been adopted an incremental development methodology, but turned to smaller equipment.

It also was used sequence diagrams, quite common in programs development and data flow diagrams, more used in projects with microcontrollers.

In Figure 3 shows a representation of the equipments used in the system development, as well as the physical interconnection between them. In this assembly, it was used an outside line Public Switched Telephone Network (PSTN) connected to the TLM to make external calls.

NodeBuilder and LonMaker softwares were used for the program development. Both are provided by Echelon. LonTalk Protocol Analyzer (LPA) software, developed by Loytec, was also used (LOYTEC, 2008).

The TLM and DTM devices were developed with NeuronChip microcontroller, and meet the following project requirements:

- to use of the smallest number of entries in the tables used for devices configuration: the device aims to keep compatible to be applied to other control network features.
- to use a telephony compatible time to the bindings configuration (or closing the phone link): avoid discomfort to the user, and maintain compatibility with already installed PABX systems.
- to dispense human operation and maintain the device full independence: ensures that the system does not require constant maintenance and keep their integrity for long periods.

As a way to evaluate the solution proposed, it was used the LonWorks® network analyzer to measure the total time consumed by the dynamic binding process proposed, comparing with the standard binding process, and checking it the result meets the time requirements of the project, that is one second (to meet the standards of Brazilian telephone).

Several tests were also conducted, involving external calls request and calls receiving, forcing the process to be performed many times.

3. PROPOSED MODEL

As a way of meeting the cited specifications in the previous item, it was added to each device an extra software layer that includes the functionality of the device, and that represents the reactive agent, as illustrated in Figure 4:
To facilitate the solution development, the problem was divided into three stages: the first one includes the handshake and the decision parameters to be used in the binding’s configuration, the second is to achieve the proper bindings, and the third stage involves the process for undo the bindings, leaving the devices available for new connections.

3.1 Stage 1

To perform this stage, the device must be in an initial state of waiting, where the agent is active and attentive to any requests for connections from other agents.

When the device receive a call or request for assistance, the agents will exchange direct messages between themselves, adjusting the parameters needed to achieve the bindings.

This stage is completed when they both have all the settings necessary for that the binding can be achieved.

3.2 Stage 2

With the parameters adjusted, they should begin the configuration process. At this point, it is worth noting that each agent is responsible for conducting its own configuration process, not interfering in the process of the another agent.

If there is any failure in the execution of this stage, the device sends a message of failure for the other agent, indicating that the process should be canceled.

After the setup process, they perform a check to ensure it was successful and that the communication can be initiated.

With a positive response of that the final assessment, it is initiated the exchange messages of telephony including the whole call progress process and voice messages exchange. At this time the agent get away from control, passing it to the basic operating software of the device.

3.3 Stage 3

When the call is finished, the agent takes back the control and performs all the configuration of the device, so that the bindings are undone and the device is ready to perform a new call.

At this stage it is not necessary the communication between agents, because it involves only the configuration of the internal tables of each device.

4. RESULTS ANALYSIS

The stage 1 obtained average time was approximately 23 ms. This time is due to the need of several messages exchanges between the devices for the parameters to get adjusted.

Furthermore, depending on the case, it is necessary that the settings table get swept several times, until the agents enter into a consensus that witch values can be used.

The traditional method of conducting bindings, made from the LonMaker tool is naturally much faster, since it does not require the exchange of messages between devices and the configuration tool already knows the free values that can be used.
It was also performed various failures simulations, such as loss of communication during the process, and all tests showed that the solution is effective in the failure recovering and left the devices ready for another attempt.

In stage 2, the average time needed to perform the required configurations was approximately 602 ms. This time was measured from the receipt of the message containing the configuration values adjusted between the devices and the first call progress message.

This time is justified by the fact that the device must to make all the configuration parameters in these tables that are stored in EEPROM memory area, a memory type that demand a longer time for completion of records.

When performing the same procedure with the LonMaker, the average time was obtained was 1.983 ms. This big time is due to the fact that besides having to write data into EEPROM memory, all the configuration and verification is done through the control network, with devices management messages.

Finally, the results analysis for stage 3, did not take consideration the completion time, but if the devices were in a state of waiting, as specified before.

After this step, you can verify that the bindings were completely undone and the devices were free to receive new connections. Figure 5 shows the devices tables at the end of stage 2.

As seen the tables containing binding information between the device TLM (MLT) and DTM (MTD), show that the telephone link between them was done.

Figure 6 shows the same table at the end of stage 3.

As seen all the configuration entries have been deleted and, therefore, the devices are free for new connections.

5. CONCLUSION

The total mean obtained in the completing the telephone link process, from the beginning of the stage 1 to the end of the stage 2, was approximately 625 ms, which is considered a good result for the telephone systems. This result means that after the user enters a phone number or make a request for external line, he will hear the calling signal or external line tone signal after about 625 ms, time barely noticeable by the user.

At the end of stage 3, the devices had been effective in returning to the original state and be ready to receive new calls.

From the point of view of multi-agents systems, it was applied an reactive agent with reservations to
the used communication means

A characteristic of reactive agents is to have a primitive method of communication, often using only the very environment in which they are inserted (WOODRIDGE; JENNINGS, 2002). In this solution, the agents exchanged messages by a complex network protocol.

However, these agents did not have any model type: of the environment model, of other agents, and of their own model. They were not able to make inferences, to perform learning or to seek knowledge.

They only reacted to external stimuli as a previously implemented logic that could not be changed at runtime, which are features of a reactive agent (WOODRIDGE; JENNINGS, 2002).

Through the application agents’ concepts, the presented solution meets the requirement of maintaining the devices autonomy, since they do not depend on external commands to configure themselves.

Also, it was not required any human intervention during the process execution, any form of communication used during the exchange of voice messages is LonWorks® standard and it was not add any overhead to the protocol. It was used only the minimum necessary parameters for the bindings configuration.

Thus, the conclusion is that all project requirements were met.

As an improvement to consider there is the NeuronChip memory implementation. As LonWorks® technology considers that bindings are permanent, theirs settings are stored in an EEPROM area, which may support approximately 1,000,000 recordings (MOTOROLA, 1997). For example, assuming that under heavy use it is made 50 calls per day from certain branch, which corresponds to 100 recording procedures in the EEPROM memory, the durability of the equipment would be approximately 27 years.

6. FUTURE WORK

In the context of the DSPTS project, one can imagine the possibility of applying cognitive agents to solve greater complexity problems and that may require a system with greater integrity and decision capability.

As example of such problems there is the decision of which attendant branch will attend to an incoming call for an installation where there are many attendants’ branches. This could be solved through the use of auctions with pre-established metrics, method widely used in societies of cognitive agents (BENISCH et al., 2004).

For a broader case that involves the entire LonWorks® technology, it can propose the implementation and evaluation of new hardware for the network nodes, enabling the storage of bindings’ information in RAM. This would remove the restrictions associated with the EEPROM memory, allowing faster bindings and a greater system life time.

A natural DSPTS continuity suggestion is related to the design of building automation systems based on LonWorks® technology that integrates the PABX functions.

7. ACKNOWLEDGEMENTS

The authors wish to thank the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), project number 05/56045-6, and the company Conceito Tecnologia, for the support conceded for this research.

8. REFERENCES


LONMARK INTERNATIONAL. Site of standardized organ for LonWorks technology and products certification. Available at: <http://www.lonmark.org/>.


