Integration of Smart Cards into Automation Networks

José Alberto Vigo Segura, Juan Jesús Sánchez Sánchez, Natividad Martínez Madrid, Ralf Seepold

Departamento de Ingeniería Telemática,
Universidad Carlos III de Madrid,
Av. de la Universidad, 30
28911 Leganés, Madrid, Spain
{jvigo,jjsanchez,nati,ralf}@it.uc3m.es

Abstract — *Individual user profiles can have strong influence on basic setup and maintenance of intelligent environments. Independently from architecture and technology, end-users' preferences need to be preserved, retrieved upon access and adapted over the life cycle of the profiles. The presented model proposes an integration of smart cards into a control network structure that can be used to setup, store and reconfigure a well-defined cluster of the network that depends on the environment in which the smart card is used. As a result, users can share and transfer profiles in different environments without any need to access a centralized repository. The architecture developed integrates a smart card as an additional device into the network and thus it allows access to efficiently transfer data between the card and the network.*

1 Introduction

The aim of this work is to present a new approach for a high-grade integration between automation networks and smart cards. The implementation of the proposed architecture will lead to an automation environment whose configuration for each user depends on the information stored in his or her card. This means that a user will be able to configure and control devices that are connected to the network with the help of a smart card. Many different scenarios are possible; a user can control from its card the power of the home lights, the maximum and the minimum temperature that its central heating shall provide, how the blinders shall behave, preferences in the TV program, etc. The proposed solution is based on a specific type of control networks, LonWorks, which provides a distributed, powerful and open architecture in order to control and manage any kind of sensors and actuators.

The structure of this paper is the following: In Section 2, both Smart Card and LonWorks technologies are reviewed since they played a central role. Besides that, a review of related work is presented. Section 3 describes the proposed architecture while Section 4 is dedicated to the definition of user profiles. Finally, the conclusions are presented.

This work has been partly supported by the Spanish Ministry of Industry under the Trust-eS (FIT-070000-2003-930) and Everyware projects (TIC2003-08995-C02-01), and by the European Commission (VI Framework Program) under Ubisec project (IST-2002-506926).

2 State of the Art

This section reviews the main aspects of smart card technologies, and further presents briefly the LonWorks network, which has been selected as automation technology. Finally, some related work is summarized.

2.1 Smart Cards

A smart card is able to store and process data and to protect the stored information from unauthorized access or tampering. A smart card reader device or CAD (Card Acceptance Device) is needed to provide power supply and a communication interface to the smart card. In this way, the smart card can operate and communicate with the external world.

The mechanism to exchange information between a smart card and a reader is defined in the ISO 7816-4 [1] standard and it is based on the use of APDUs (Application Protocol Data Units), which are information packages with a specific format. A smart card never initiates the communication with the CAD but it answers to commands sent by the reader. Two kinds of APDU are defined, the COMMAND APDU (sent by the CAD to the card) and the RESPONSE APDU (sent by the card as an answer to a COMMAND APDU) (see Tables 1 and 2).

Code	Name	Length	Description
CLA	Class	1	Class of instruction
INS	Instruction	1	Instruction code
P1	Parameter 1	1	Instruction parameter 1
P2	Parameter 2	1	Instruction parameter 2
Lc	Length	variable	Number of bytes of command data field
Data	Data	Lc	Bytes sent in the data field of the command
Le	Length	variable	Max. of bytes expected in the response

Table 1: Command APDU contents

Code	Name	Length	Description
Data field	Data	variable=Lr	Data field of the response
SW1	Status byte 1	1	Command processing status
SW2	Status byte 2	1	Command processing qualifier

Table 2: Response APDU contents

2.2 LonWorks Networks

The LonWorks network [2] differs form other control networks in the fact that it develops a decentralized control network instead of the traditional automation network with a central control element. In order to ensure the interoperability between devices from different manufacturers, LonMark association ([3], [4]) was created with the mission of developing standard profiles to ensure this interoperability by the certification of LonWorks products and promotion of the benefits of interoperable systems.

The architecture of a LonWorks network consists of several intelligent devices or nodes that communicate with each other using a common protocol (LonTalk protocol [5]) over different communications channels. Each particular node is associated with a specific functionality in the network and the results can be shared with any other node installed. The LonTalk protocol defines two ways to exchange messages between the different devices. The first one is based on sending and receiving network variables that are static objects (input or output), they represent a standard way to send and receive information from and to other devices. The second way is based on explicit messages, however this not implemented by the majority of the manufacturers.

Every LonWorks network needs a process in which a specific functionality is associated to each device on the network. This is achieved by programming the device with an application that can be divided into one or more functional blocks. Each of these blocks is the implementation of a specific functional profile that can be considered as a template for the achievement of the functional blocks. A functional profile is actually a set of network variables and configuration properties, which together defined a specific behaviour for a device.

The next step involves the assignment of network variables to nodes, this is known as *binding*. Through this process, a virtual wire is established between LonWorks devices, so that, when a network variable changes its state or value, the new value is transmitted to all bound devices.

The last step of the deployment process involves the creation of an external interface for each device [3], through which the rest of LonWorks nodes know the function of this device and can communicate with it. The external interface of a LonWorks device is the network visible interface to a device. It is comprised of the elements shown in Table 3.

Parameter	Functionality
Neuron ID	A 48-bit unique identifier for a LONWORKS device.
Standard program ID (SPID)	Number for identifying the device interface.
Device channel ID	Optional number to specify the channel to which the device is attached.
Device location field	Optional string or number, which specifies the device location.
Device self-documentation string	A string that specifies the functional blocks on a device.
Device configuration properties	Configuration data used to configure the device. Functional blocks may also have configuration properties.
Functional blocks	Logical components implemented on the device. There is usually a list with the implemented network variables.

Table 3. Elements of external interface

Different attributes can be added to the network variables in order to provide additional features. So, network variables can be established as authenticated (authenticated messages are used to transmit their values), with priority (meaning priority time slots are used to transmit their values) and synchronous (all the values assigned to the network variable are propagated). Finally, network variables (together with some configuration properties) are generally grouped in functional profiles in order to perform a single function in a device.

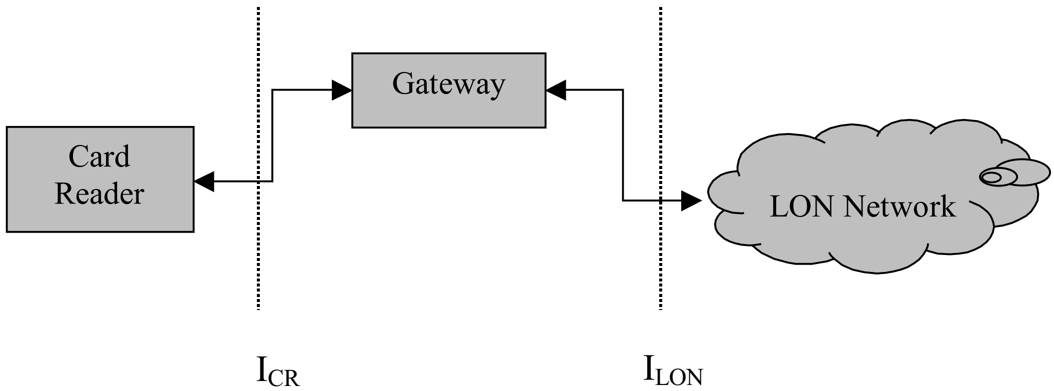
2.3 Related Work

Until now, interactions between automation networks (both for an industrial or office/home environment) and smart cards are dedicated to user authentication. Examples can be found in the research projects FUTURE HOME [6] or ePerSpace [7]. A more similar conception can be found in a SUN white paper [8] where a wider vision of a customisation of environment with user profiles is presented as a possible scenario. This paper describes how will be possible to use user profiles in a "connected home" in order to configure the network according to users' preferences. However, it is not a technical proposal but an example to support the idea of home networks based on a gateway.

3 Interoperability proposal

The purpose of this paper is to introduce an architecture (see Figure 1) to configure a control network, in particular a LonWorks network, from a smart card. The key element of this design will be a gateway. The gateway is a dedicated LonWorks node that acts as bridge between the LonWorks network and a standard card reader. It is responsible of interpreting the information stored in the card and configuring the state of the network depending on that information. Furthermore, it must be able to program the card from the network configuration associated with the environment where it is installed.

A control network is designed in order to maintain a functionality that does not change over a large period of time once it is installed and configured. This idea common to any control network and LonWorks inherits this feature. As a result, there are two different phases: the installation phase and the operation phase. In the installation phase the network is installed and configured, while in the operation phase, the network process the configuration defined, i.e. action according to its configuration when an event happens in the network.



I_{CR}: Card Reader interface, I_{LON}: LonWorks network interface

Figure 1: Architecture to integrate a Smart Card in a LonWorks network

This phase forces the network to interact. A gateway must provide a two functionalities: programming a new card with the profiles stored in the network (Configuration Mode) and, on the other hand, receiving information from a card programmed previously and then configuring the network (Operation Mode).

In order to control the home lights, for example, the LonWorks network has implemented some standard LonMark Profiles, such as the Light Sensor, Switch, Manual Override Switch, Constant Light Controller and Lamp Profiles. These profiles define both the set of network variables and the relationship between them in order to perform a concrete control activity. So, through these profiles, it is possible to control the lighting of a home both in a manual way and through an illumination level setpoint.

A simple example of customizing this network could be a smart card where the user has stored its personal configuration for the level of illumination. With the help of the card, the user could fix both the illumination setpoint and the state of the different lamps. This means configuring two concrete networks variables for each element to control: the network variable which selects the operating mode and adjusts the setpoint of the constant light controller (*nviSetting)* and the configuration property that changes the illumination level setpoint for the controller (*nciLuxSetPoint*).

Therefore, the gateway has to be programmed with variables of the same during the process of installation of the LonWorks network. Finally, they need to be bound them to the node or the nodes which implement the Constant Light Profile. This means that in the Configuration Mode, the gateway will send to the card both the network variables and the value associated to them in that moment.

Another important aspect must be considered: the format of the information to be stored in the card and to exchange between the network and the card. In a first approach the format stored in the card is directly used by a LonWorks network.

Along the following subsections, different logical structures used to store network information are explained.

3.1 Interoperability proposal

The proposed solution is an intermediate step between an independent structure and a structure completely dependent on the control network used. Thus, the smart card will store data structures that refer to a concrete network and gather a set of registers through which the network configuration is established. Each data structure is referenced by an identifier that must uniquely represent the control network; this environment identifier is described in the next subsection.

Each register contains the name of a network element to configure, its value and information about its validity period. In this approach, the network variable to control and its value in Neuron C [8] (the programming language for LonWork devices) format will be stored in name and value fields. Therefore, there is an intermediate dependence between information stored in the card and the control network since general data structure for storing information about control networks has been created but it has been applied to a concrete control network.

3.2 Environment identifier

The Environment identifier identifies the LonWorks network and acts as an index to the data structure. In order to find out which is the identifier of the network, the gateway has to extract this information from the network. Thus, each LonWorks device has to incorporate a network variable for identifying its environment. The gateway will ask for the identifier of all nodes installed in the network and store it in memory. When the card is inserted in the card reader the gateway has to send the identifier to the smart card.

3.3 Smart Card and Gateway Communication

In order to provide the management functionalities assigned to the smart card, a specific application, called CMA (Configuration Manager Application), has to communicate with the LonWorks network. This application manages the different user configurations.

When the card is inserted, the gateway has to select the right application to operate with the card (one card can store different applications). Once the CMA application is successfully selected, it will be possible to establish a communication between the program running in the gateway and such an application. In order to select the CMA application in the card, the following APDU should be sent by the gateway (Table 4):

CLA	INS	P1	P2	Length	Body
00	A4	04	00	AID Length (Hex)	AID

Table 4: Profile Manager Application Select APDU

The APDU asks the card to load a certain application identified by its AID that is a bytes sequence whose length can vary between 5 and 16 long that uniquely identifies a smart card application and it is assigned by ISO. The expected response to the SELECT Application command is a 0x9000 APDU response when the selection is successful; in any other case an error APDU will be sent by the Smart Card (Table 5).

SW1	SW2	Data field
90	00	--

Table 5: OK response APDU

After successfully selecting the CMA application, the first step for the gateway is to send the environment identifier associated with the network to the CMA. With this purpose the gateway uses the corresponding APDU command shown in Table 6.

CLA	INS	P1	P2	Length	Body	Le
AA	01	0?	00	10	Environment ID	--

Table 6: APDU used to send the Environment ID to the CMA

Again, the normal response to this command is a 0x9000 APDU response if the selection is successful; in any other case an error APDUs will be sent by the smart card (any different value). Depending on the gateway mode (see section 4) this APDU will force the CMA to create a user configuration inside the card where network variables will be stored or it will cause that the network variables associated with the chosen environment are required.

4 Gateway Operation Modes

This section describes in detail how a smart card and a gateway interact in both gateways modes: Configuration Mode and Operation Mode. The Configuration Mode is necessary in order to provide to the card the set of customized profiles to control and configure the network. In the Operation Mode, the gateway will operate as a "translator" between the card and the network.

4.1 Configuration Mode

In this mode, the gateway sends to the smart card the current configuration of the network. Thus, the gateway must send to the card both the environment identifier and all the information about its external interface. This point is highly dependent on the way that the information will be stored in the smart card (SC in figures) or in the gateway. In principle, this approach considers that the card is going to store all pairs of network variables and values.

In particular the set of network variables that are bound to the rest of the devices on the network must be packed into APDUs. In case of a network variable array, each element will be sent as an individual network variable, with the number of the element in the array as a part of the name field.

When a card is inserted into the card reader, the gateway selects the CMA program and then it sends the identifier of the LonWorks network to the card. Through this identifier, the card is able to create a data structure to store the configuration of the network. Then, the gateway sends all the Network Variables (NVs) defined in its external interface to the card.

So, in the previous example where the illumination was controlled by the card, the gateway has to send to the card two variables for each device or a set of devices under its control: the network variable *nviSetting* and the configuration property *nciLuxSetPoint* with their current values. It is important to remark that previous to the sending, the network and the devices must be configured in the state the user wants because these values will be the ones that the card stores.

All transactions are confirmed, so the CMA will response with an OK response APDU whether it is able to allocate the information the gateway is sending. At the end of this sequence, the smart card will have received and stored all the available information from the gateway for configuring and controlling the LonWorks network (see Figure 2).

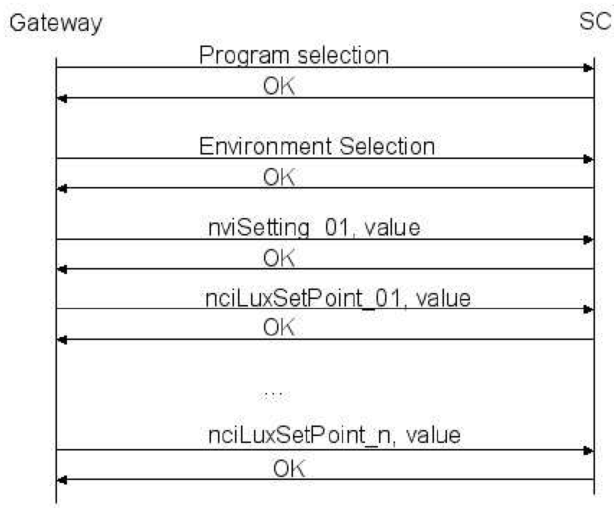


Figure 2: Sequence of messages of the Configuration Mode

The card will store the received variable unless an error occurs; in such case, an error response will be sent to the gateway that it will stop the communication. Thus, the transaction of configuration between the gateway and the card will be incomplete. In general, three possible errors can be identified:

- Application Error. The error is produced due to a wrong selection of the application in the card either because the application does not exist or the application is not prepared to talk to the gateway.
- Environment Error. This error is produced because the application cannot create the structure to store the data or a structure with the same identifier exits in the card. In case a user wants to reprogram a configured card for a network in the same network, it is necessary that the data structure associated with this environment has been deleted.
- Request Error. This error occurs due to a lack of memory in the card.

4.2 Operation Mode

When the gateway is running in this mode, it is ready to receive instructions from the card in order to configure the LonWorks network. The gateway acts as a server that receives data stored in card; it processes them and updates the network.

The process of network configuration starts when a card is inserted in the card reader and this is depicted in Figure 3. Once the CMA receives the request for sending the data structured associated with the network environment, it sends a response APDU with all the information associated with each variable, including its stored value, for each request. Responses will have the SW1 field set to 0x91 if there are left variables and it is set to 0x90 after the last one. Therefore, it will be possible for the application in the gateway to notice that the communication of variables is over. If there are no variables associated with such an environment, an error APDU will be sent to the gateway.

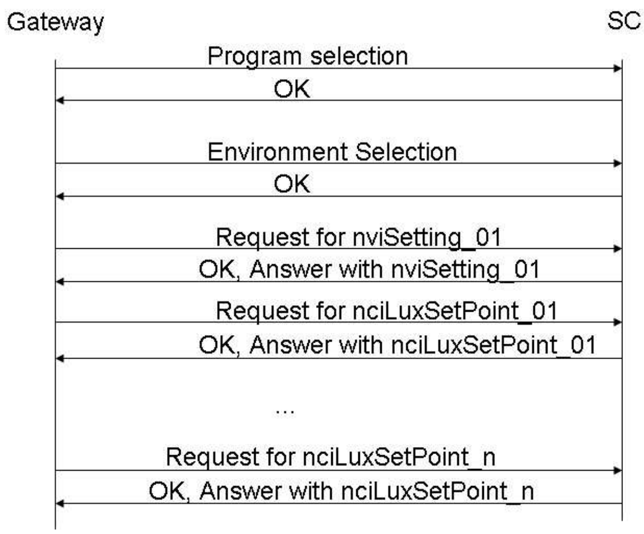


Figure 3: Operation Mode Scenario 1 Sequence of messages

Once the gateway receives an APDU from the card, it has to check that the referenced network element in the APDU is previously declared in its external interface. Otherwise it must discard that APDU and it stops communication.

In order to configure the illumination, in the initial example, the user has to send the set of network variables stored for this concrete network. This means sending the variables and the values stored in the previous Configuration Mode.

5 Conclusions and Future Work

The proposed architecture aims to realise a first approach to an intelligent environment able to adapt to the preferences of a user. Such preferences will be stored in a personal card and they will be used to configure the network when the card is inserted in a reader (or when a contactless card passes close to a contactless reader).

In the next phase of the development, the creation of enhanced user configuration structures will be undertaken with the purpose of extending the degree of functionality of the network. Also a study will be developed, measuring the influence of operations and of the performance achieved when more management operations are assigned to the smart card. Furthermore, it will be possible to make an evaluation of the advantages and drawbacks of distributing the computational cost between the smart card and the gateway. Other future studies will address the integration into a residential gateway like OSGi [10] gateways.

References

[1] International Organisation for Standardisation (ISO), JTC 1/SC 17. ISO/IEC 7816. Identification cards - Integrated circuit(s) cards with contacts, 1995.

[2] Echelon Corporation. Introduction to the LonWorks System, 1999. Available at http://www.echelon.com/support/documentation/manuals/078-0183-01A.pdf

[3] LonMark Association. LONMARK Application-Layer Interoperability Guidelines version 3.3, 2002. Available at http://www.lonmark.org/products/guides.htm

[4] LonMark Association. LONMARK Layer 1-6 Interoperability Guidelines version 3.3, 2002. Available at http://www.lonmark.org/press/download/LmPhy33.pdf

[5] Echelon. ANSI/EIA709 Control Network Protocol Specification (LonTalk). (CEA-709.1). 2002. http://www.echelon.com/products/lonworks/protocol

[6] Juha Koivisto. Wireless Home Network Architecture and Concepts for User Interactions". Technical Report of the Project FUTURE HOME. (IST 2000-28133), 2000

[7] WP3 contributors, G. Nygreen, P. Plaza. Detailed workplan and specification for intermediate test beds. Technical Report of the Project ePerSpace – IST Integrated Project (N° 506775), 2004

[8] SUN Microsystems, The Connected Home Powered by Java EmbeddedServer™. 2001.http://java.sun.com/products/consumerembedded/whitepapers/dot.com.home.pdf

[9] Echelon Corporation. Neuron C Reference Guide Revision 5, 2003. http://www.echelon.com/support/documentation/manuals/078-0140-02E.pdf

[10] The Open Services Gateway Initiative. OSGi Service Platform Release 3 March 2003OSGi Service Platform Release 3. 2003. Available at OSGi Official Web Site. www.osgi.org