

# Controlling home appliances using XML

Gantumur Natsag<sup>o</sup>, Dong-Kyun Kim, Byung-Chan Jeon\*, Sang-Jeong Lee  
Dept. of Information Technology Engineering, Soonchunhyang Univ  
Div. of Computer Science, Chungwoon University\*

## Abstracts

Recently, as many next-generation appliances are become digital and intelligent, companies have come out home appliances to communicate with each other through home network. Therefore, it requires various kind of communication technology to control home appliances.

In this paper, XML messages which can control home appliances such as boilers, TV etc. are proposed and defined. The XML messages sending from a home server to home appliances should be fully defined all needed information to activate and deactivate home appliances.

## 1. Introduction

Present time most of homes are equipped with home appliances. The rapid development of personal computers and communication resources are supporting to progress more advanced home networking. This kind of situation requires smart devices have to operate through any kind of communication technology to home appliances. Definition of home networking is the collection of elements that process, manage, transport, and store information, enabling the connection and integration of multiple computing, control, monitoring, and communication devices in the home [1].

XML's hierarchical structure of data representing expresses meaning of data, infinite tag allows customization of building structures [4]. This kind of ability in XML makes possibility to use at processing information between home appliances in home network [5]. Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language [2]. The Document Object Model (DOM) provides a standardized way to access and manipulate the information stored in XML documents. DOM application programming interfaces (APIs) serve as a bridge between applications and XML documents. The Document Object Model (DOM) presents an easily processed standardized interpretation of an XML document to applications and scripts [3].

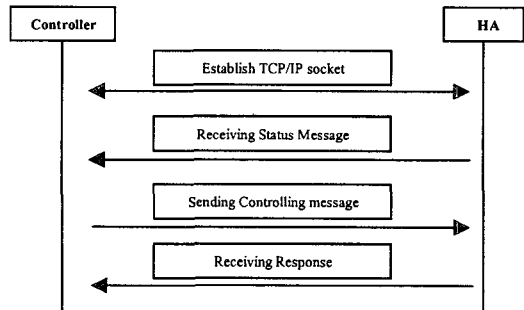
In this paper, XML messages which can control home appliances such as boilers, TV etc. are proposed and defined. The XML messages sending from a home server to home appliances should be fully defined all needed information to activate and deactivate home appliances. Controller application written in c++ language creates XML document

parsed by DOM (Document Object Model) and sending it through TCP/IP protocol to control home appliances. So as a home appliance we used simulator able to operate as home appliance, that written using Flash MX.

## 2. Communication Model

Software components on various computers can communicate to each other over network in many ways. One of technique is using transport layer's call interface, such as socket mechanism [6]. So communication can be establishes through custom communication protocol. We used the TCP/IP socket communication model to establish connection from Controller to home appliance.

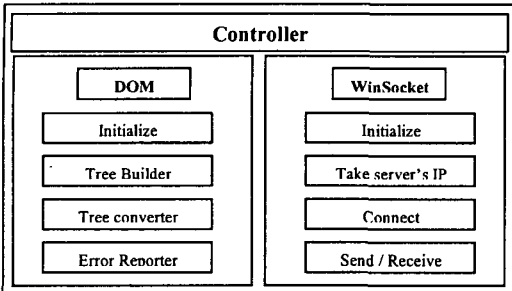
In a Figure 1, it shows the sequence of establishing communication, receiving home appliance's status message, sending Controller's message and response from home appliance.



[Figure 1] Control Flow Diagram

Controller mentioned here is consists of two parts. These parts are XML DOM parser and WinSocket communication

protocol. XML DOM parser creates XML messages using DOM implementation and sending it through Winsock communication protocol.



[Figure 2] Controller Structure

### DOM Parser

**Initialize** – Creates MS-XML DOM Instance object

**Tree Builder functions** – Includes Creating elements, Adding elements as a child

**Tree converter** – Converting created DOM object to string to send through socket

**Error Reporter** – Reporting error messages while working with DOM Instance object

### Winsocket

**Initialize** - Creating Socket

**Server IP** – Takes IP address of home appliance

**Connect** – Connect to home appliance

**Send** - Sending XML string through established socket

**Receive** - Receiving incoming XML data

Structure of home appliance is almost the same as Controller. In our case home appliance is Flash simulator. It uses XML\_Socket to accept connection from Controller then using Flash XML parser to read XML message from Controller and parse it. Actions take depending on parsed message and send back status message.

### 3. Defining XML message

As a mentioned above XML message going through Socket communication channel between Controller and home appliance need to be several to express message's role. For example we can say Controller is establishing connection to Boiler.

1. Controller receives Boiler's status message.

2. Send Controlling message to Boiler depending on message received from Boiler.
3. Boiler receives message and does activation, deactivation or do some actions depending on received message.
4. Boiler sends back message that has status information of Boiler.

Using XML to define the syntax of Controlling and Response messages are simple. After Controller connected to home appliance, home appliance sends status message back to Controller.

The Controlling message "Turn On Boiler" to home appliance (Boiler) is shown in Figure 3.

```
<?xml version="1.0"?>
<Body>
  <Device>boiler</Device>
  <Control>
    <Power>on</Power>
  </Control>
</Body>
```

[Figure 3] Turn On Boiler

Also Controller can send another message that does have a field SetHeatingTemp and 70 as value. This message means Controller wants to set heating temperature to 70 degree.

```
<?xml version="1.0"?>
<Body>
  <Device>boiler</Device>
  <Control>
    <SetHeatingTemp>70</SetHeatingTemp>
  </Control>
</Body>
```

[Figure 4] Setting Heating Temperature

After Boiler received controlling requests then does related actions due to message and sends back status message. The message in Figure 5 is send back message from Boiler after message in Figure 4 received from Controller.

```
<?xml version="1.0"?>
<Body>
  <Device>boiler</Device>
  <Status>
    <Power>on</Power>
    <SetHeatingTemp>70</SetHeatingTemp>
    <CurrentHeatingTemp>65</CurrentHeatingTemp>
  </Status>
</Body>
```

[Figure 5] Boiler status message

These three messages are basic describing the basic actions.

#### 4. Implementation

Controller is written by Visual C++. It used Winsock2.0 API and MSXML4.0 API. Application classes consist of two parts XML parser classes and Socket communication classes.

##### 4.1 Classes

First Document Object Interface needs to be initialized.

```
IXMLDOMDocument * DomFromCOM()
// Creating DOM instance
```

[Figure 6] DOM initializing function

After initialized the DOM instance all elements of XML message are created in instance. While adding elements to the instance we used user defined functions. Such as allocating variant string, add white spaces, report error if occurred, converter converts DOM tree structure to string to send through socket interface. These functions are shown in Figure 7.

```
VARIANT VariantString(BSTR str)
//Allocating memory for DOM tree
Void AddWhiteSpaceToNode( IXMLDOMDocument*
pDom, BSTR bstrWs, IXMLDOMNode *pNode)
//Adding white space after inserting a new node
Void ReportParseError( IXMLDOMDocument *pDom,
char *desc)
//detects errors during the creating and parsing XML
Void dprintf( char * format, ...)
//converting DOM tree to string
```

[Figure 7] DOM functions

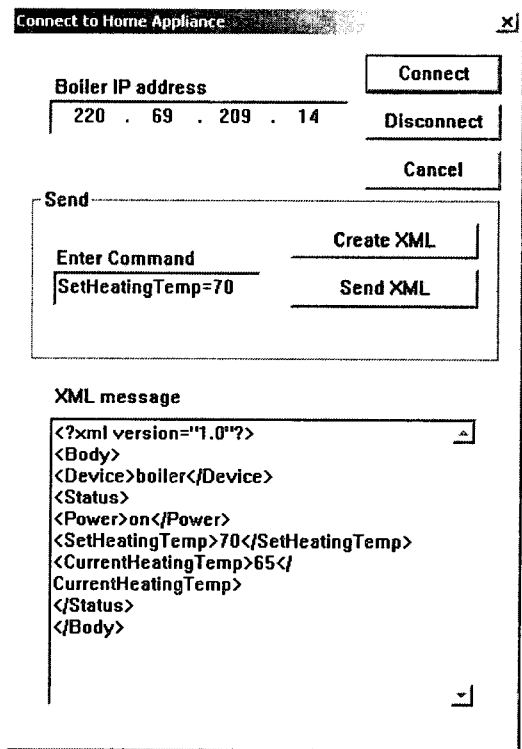
As mentioned before both side, Controller and Boiler needs to Initialize Socket channel. Socket communication mechanism is the most common to public. So doesn't need to explain fully. Down bellows are user defined functions to take action on Controller using Socket API.

```
SOCKET NetworkStart(HWND &_hDlg)
//socket Initialize
void SendText( SOCKET &_sock , HWND &_hDlg)
//send XML message
void NetworkEnd( SOCKET &_sock )
//Disconnect from server
```

[Figure 8] Network functions

##### 4.2 User Interface

Figure 9 showed the User Interface of Controller. User enters IP address of home appliance in 'Boiler IP address' field, then click 'Connect'. This will initialize connection with Boiler and receives Status message from Boiler. Received XML message will be shown in XML message field. After user looked at received message, user will decide to enter either 'On' or desired heating temperature such as '70'. Create XML button will create XML message depending on user's inputted value. After XML message created, it will shown in 'XML message' list box, then User can check it and can send to Boiler by clicking 'Send XML' button. When Controller sends 'Controlling' message, then Boiler receives it and does some related actions by looking at received message. At the end Boiler send back Status message that describes Boiler's current condition.



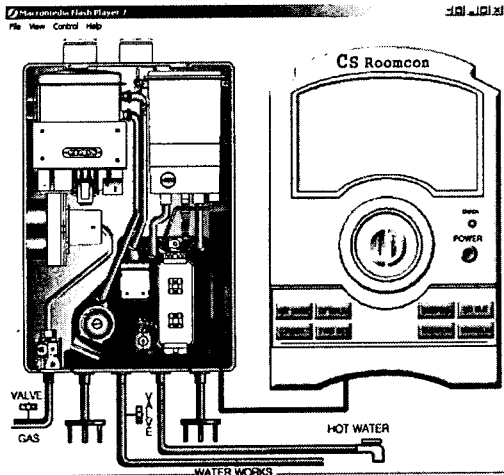
[Figure 9] Controller

Figure 10 shows Boiler Simulator's off status before receives connect request. Boiler Simulator is kind of Server application and is listens connection.

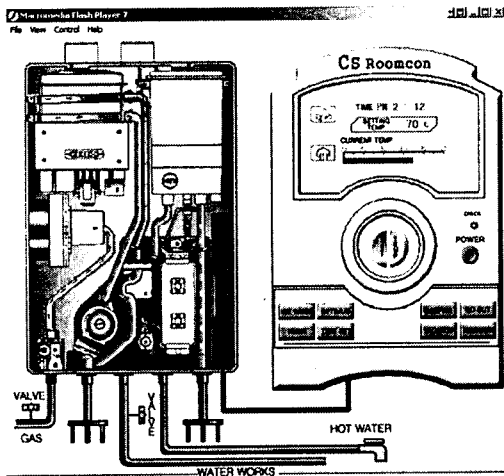
Figure 11 showed home appliance simulator after established connection requested by Controller, Boiler sends status message to Controller and receives XML message, then parse it using XML parser classes. Boiler executes one

of the action turn on or off Boiler, Set heating temperature depending on parsed values.

message flows. Later paper will be extended to design and implement the concept UPnP.



[Figure 10] Simulator Off



[Figure 11] Simulator On

## Reference

- [1] Web Proforum Tutorials  
"Home Networking" <http://www.iec.org>.
- [2] Extensible Markup Language (XML) 1.0  
(Second Edition) W3C Recommendation  
<http://www.w3.org/TR/2000/REC-xml-20001006>  
6 October 2000
- [3] Microsoft Developer Network  
"Processing XML with MSXML"  
<http://msdn.microsoft.com/XML/BuildingXML/ProcessXMLwMSXML/default.aspx>
- [4] Antonio Kung, Barbara Raither, Steve McConnell  
"Electronic Commerce Services Expand Home Automation Capabilities"  
EMMSEC 99 Conference June 1999
- [5] Friedemann Mattern, Peter Sturm  
"From Distributed Systems to Ubiquitous Computing"
- [6] Jeffrey Nichols and Brad A. Myers.  
"Studying The Use of Handhelds to Control Smart Appliances". International Workshop on Smart Appliances and Wearable Computing. IWSAWC 2003. In the Proceedings of the 23rd IEEE Conference on Distributed Computing Systems Workshops

## Conclusion

This paper implemented XML messages sending through socket communication layer between Controller and home appliance simulator. They are consisted of Status, Controlling and Response messages. It is not enough to express all functionality of home appliance by 3 kinds of