Remote Controlling Devices Using Instant Messaging

Building an Intelligent Gateway in Erlang/OTP

Author:

Simon Aurell

IT University of Gothenburg

Box 8718, 402 75

Gothenburg, Sweden

simon@erlang-consulting.com

Examiner:

Annika Aasa

IT University of Gothenburg

Supervisors:

Marcus Taylor

Erlang Training and Consulting Ltd.

Francesco Cesarini

Erlang Training and Consulting Ltd.

ABSTRACT

This paper shows how instant messaging technology can be used for remote controlling of devices, and outlines some of the issues involved, of which the most important is security. The concept of controlling and monitoring devices using instant messaging dialogue, presence and buddy list features is applied to a home automation context, and the idea and implementation of a prototype system is described. The paper describes how the excellent robustness and prototyping qualities of Erlang/OTP were exploited to quickly build a prototype system. It also shows how a gateway capable of speaking multiple device protocols can provide a single access point to different kinds of devices and services, and how the concept of agents can be used to add a layer of intelligence to the set of devices being controlled or monitored.

Keywords

Home Automation, Instant Messaging, Erlang, Agents, Networked Appliances, Remote Controlling, Ubiquity, Presence

1. INTRODUCTION

Home automation technology, although not yet as widely adopted in Europe as in the US, is becoming more and more available, affordable and interesting to the wider public. What once was not far from novelty products, like the "Clapper" that switches the lights on by the clap of your hands, is now becoming an important factor of utility, convenience and usability in modern houses. These factors are added to the already substantial possibilities of entertainment and security created by home automation. Accessibility aspects of home automation also become more interesting, considering the current demographical shift in the western world, which will create an increased demand for accessibility in homes and possibly for related facilities such as remote health monitoring.

The next important step to leverage this technology is to expand and improve the ways of accessing and using it. A vital part of this is to enable remote access and location independence.

Instant messaging protocols are today used for human to human interaction. However, the concept could be extended for use in remote controlling, where it could be used to query, send commands to or receive alerts from a device. This could also be applied to services, such as a banking service. In this way, these tasks could be performed through a dialogue with a "buddy" on an instant messaging client's buddy list. The concept could also be used to mediate machine-to-machine communication.

1.1 Aim of the Paper

The aim of this paper is to show how an intelligent instant messaging based solution can be used for remote controlling of devices, and what the main issues of this concept are. The aim is also to show how it could be implemented. Therefore, the design and implementation of a prototype, where the concept is applied to home automation devices, is described. The idea behind this system is to create an intelligent, protocol independent and location independent gateway to a set of home automation devices.

The paper aims to answer the following questions:

1. What are today the main issues in the remote controlling of devices in a home automation context?
2. What are the benefits of using Instant Messaging technology for the remote controlling of devices and services?
3. How could an intelligent instant messaging based solution for remote controlling be implemented?

It also describes how Erlang/OTP [6] was used for the purpose of fault tolerance and robustness in the instant messaging server, and for quickly building a functional prototype.

This paper is intended for technical as well as academic readers. However, the description of the prototype system requires some technical knowledge. It is assumed that the reader has a basic understanding of what instant messaging is.

2. REMOTE CONTROLLING ISSUES

This section will look at the issues involved in device remote controlling today. Roychowdhury and Moyer [3] identify four "primary reasons" that network enabling of devices has not yet happened to any larger extent with the current technology:

1. Interoperability
2. Scalability
3. Security
4. Limited Services

Interoperability refers to the capability of devices of different types and from different manufacturers to communicate and cooperate. This is related to usability, since it is essential that a user does not have to be concerned with details of which protocols are used or similar technical details. *Scalability* refers primarily to scalability in terms of geographical distance and location independence. In the context of offering remote access as a service, scalability in terms of capacity would also be an issue. Roychowdhury and Moyer suggest that the issue of *Limited Services* is due to bandwidth limitations of the past, which made "a lot more concentration [go] into optimisation rather than feature provisions". *Security* is probably the most important issue, since it has implications for safety: "human lives are of concern here. By breaking into a home network, it becomes possible to control every appliance that was connected to produce disastrous results".

Security

To secure the communication between the user and the system, we need to protect the data transfer against "man in the middle" (interception and modification or eavesdropping), "replay" (messages are intercepted and later resent) and "denial of service" (attacker sends a flood of messages too large to handle) attacks.

Secure Socket Layer encryption (SSL) can be used to encrypt the data transfer and protect it against man in the middle attacks. Some instant messaging clients provide SSL today. However, SSL implies an overhead which could potentially be a performance issue.

The system would not be protected against replay attacks by SSL. The problem could however be solved by modifying the protocol to use time stamped or otherwise *uniquely identifiable messages*. This would require the client (and possibly the server) to perform additional decoding/encoding and verifying operations on messages.

Denial of service attacks would be a server issue. Since clients and agents are only listening to the server, they can be protected from flooding if server only forwards messages from trusted users.

See also a solution to security issues in a similar context outlined by Werthschulte, Westermeir and Schneider [1].

Usability

There is also an issue of usability involved in this kind of remote controlling, since it is not obvious how to represent:

1. Diverse types of communication
2. Communication with diverse types of devices
3. Feedback from and status of devices being controlled or monitored

Also, the fact that many mobile devices have small screens makes it more difficult to create good user interfaces.

3. USING INSTANT MESSAGING FOR REMOTE CONTROLLING

Roychowdhury and Moyer [3] identify four types of communication when remote controlling devices:

1. *Control* (e.g. "turn on the coffee maker")
2. *Query* (e.g. "how much milk is left in the refrigerator?")
3. *Event Notification* (e.g. "tell me when my fire alarm goes off")
4. *Multimedia Session* (e.g. "view the babysitter cam")

These types of communication, as well as the representation of feedback and status, fit very well with the concept of instant messaging dialogue and presence. Multimedia sessions might be problematic when using some clients, although this is not a limitation of the instant messaging concept in itself. Examples of instant messaging clients who do support multimedia sessions include MSN Messenger [1] and Skype [10].

Instant messaging has good scalability properties, in that it is independent of location or geographical distance, and is easy to scale up and down on server level to increase or decrease capacity.

Also, a substantial amount of reuse is possible if existing protocols, servers and clients are used. Reuse can also be achieved from a usability point of view, since many users are already familiar with the concepts and user interfaces of instant messaging. Many computers and mobile devices also already have instant messaging clients installed.

Instant messaging clients provide a good user interface for a generic representation of different kinds of communication peers, and a compact representation suitable for small screens on mobile devices.

4. USING ERLANG/OTP FOR THE PROTOTYPE

This section will explain the benefits of using Erlang/OTP when building the prototype.

4.1 What Is Erlang/OTP?

The Erlang language and the OTP [6] were developed at Ericsson for use in telephone switches with extremely high requirements on robustness and availability. It uses a high level functional-declarative syntax and is designed for massive concurrency. A fundamental concept is to model reality in processes, the obvious example being a process representing a phone call.

Although developed for telecom applications, Erlang/OTP has proven to be suitable for all types of distributed, fault tolerant, massively concurrent soft real time systems.

4.2 Why Use It for the Prototype?

Instant messaging is server based, which means requirements on robustness, availability, scalability and to some extent performance. This made Erlang/OTP a good platform for the server part of the system.

Erlang also proved an excellent language for quickly writing the rest of the prototype in. This is due to two things. Firstly, because of its high level and declarative nature, which lends itself to quickly writing powerful programs. Secondly, because of its focus on concurrency and message passing. This is very fitting, considering the central issues in this part of the system are state machines (i.e. device representations), server connections and the passing of events and commands.

5. PROTOTYPE SYSTEM

This section describes the concept and implementation of a prototype system for remote controlling devices using instant messaging.

5.1 System Concept

The idea behind the system is to create an intelligent gateway to a set of devices or services. Server based remote access to this gateway (using instant messaging) can then be offered as a service to a large number of users. By adding a "layer" of agents, we make it possible to incorporate decision making and make non-smart devices appear smart. In the prototype, this concept is applied to a home automation context.

5.1.1 Multi Protocol Gateway

The Device Connector in Figure 1 connects the automated home and its set of devices to the outside world through the agents, acting as a gateway. The gateway communicates with the devices using device drivers or some form of translators. From the agents' point of view, the translation to different device protocols is transparent. In the prototype, only support for X10[1] devices is implemented, but the gateway could easily interface to any other kind of device or service, provided it is possible to create some kind of device driver for it.

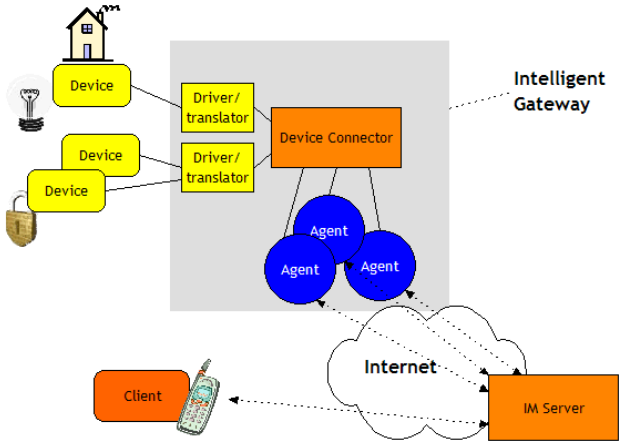

Figure 1. System overview. The intelligent gateway connects the devices to the outside world through a number of agents, which handle the dialogue with the user in the client end.

5.1.2 Agents

The agents all have their own connection to the server. This way, messages can be passed back and forth to the client. In general, the functionality of the agents is to:

1. Send commands and queries to devices
2. Forwards device alerts which the client wishes to be notified of
3. Perform tasks in response to events.
4. Perform scheduled tasks

Although the agents implemented in the prototype are not provided with a very impressive intelligence, they provide something to which one could easily attach more advanced intelligence and decision making capabilities.

[1] X10 is an inexpensive, low-end technology based on communication over the mains wiring. See [2].

One of the uses of intelligent agents is as information filters. Consider a scenario where the set of connected devices contains many sensors producing a large number of events which are not always interesting. The setup does not have to be a single home; it could be an office building, a school or a factory. An agent will monitor all these events and only report or take action on relevant or relevant combinations of, events.

It is worth noting that this architecture does not limit us to remote controlling over long distances. For example, a PDA connected to the home network could be used as a sort of universal remote control inside the home, e.g. sending a message *Turn on Channel 4* or *Play Mahler's fifth* to the "entertainment" agent on the buddy list. The agent would then figure out that the first message was meant for the TV set and the latter one should be sent to the CD player, the MP3 player, the computer or whichever other device has Mahler's fifth in store. Perhaps the user would first make the query *Have I got Mahler's fifth?* and if the answer is no, the agent could buy it from an online music store.

5.2 Gateway Implementation

The principal parts of the gateway are:

- A number of finite state machines (the devices).
- A number of intelligent servers (the agents). In other words, conceptually the agent is a server which the user sends requests to.
- An event server, which provides an event subscription/notification facility to the agents.

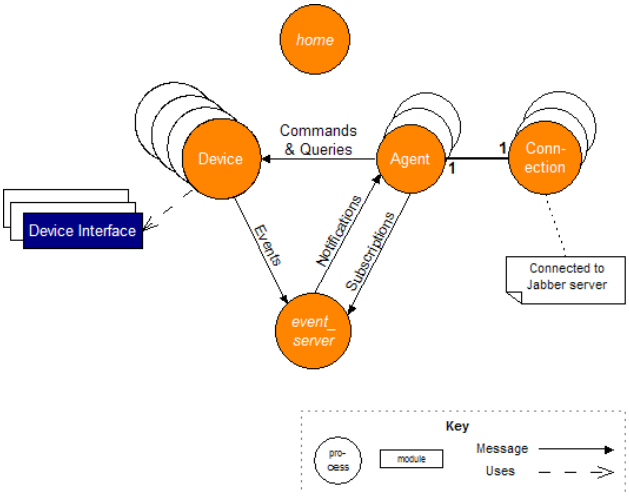

Figure 2. Key Erlang processes in the gateway

Figure 2 shows the key Erlang processes in the gateway. Note that the Agent and Connection processes have a one-to-one relationship. Supervisor structures have been left out in the figure for the sake of simplicity[2].

Table 1 shows the responsibilities of the processes in Figure 2. It also states the Erlang behaviour the processes implement.

[2] Although supervisors are a key OTP strategy for robustness, they are implemented in a fairly generic way. For more information on supervision principles in Erlang, see the Erlang/OTP documentation on the subject [6].

Table 1. Behaviours and responsibilities of processes

Process	Behaviour	Responsibility
Home	-	Creates the other processes, and then holds system state.
Device	Finite state machine (gen_fsm)	Represents a device. Uses a Device Interface to execute commands and queries. Catches events from its device and passes them on to *event_server*.
Event_server	Server (gen_server)	Lets agents subscribe to events from certain devices. Receives events from devices and distributes them to subscribers.
Agent	Server (gen_server)	Parses incoming requests from clients and/or monitors devices. Takes action based on user commands and device events.
Connection	Finite state machine (gen_fsm). Uses Jabberlang library.	Provides an agent with a streaming XML connection to the instant messaging server.

5.3 The Server

Ejabberd [15] was used for the server. Ejabberd is an open source instant messaging server written in Erlang. The clients communicate with the server using Jabber [13]. Jabber is a generic instant messaging and presence protocol based on XML streams. The Internet Engineering Task Force (IETF) has formalized its core XML streaming protocols as an approved instant messaging and presence technology under the name of Extensible Messaging and Presence Protocol (XMPP) [8].

Ejabberd does not have the feature of providing time stamped messages, but does support SSL. However, since the source code is freely available, any required security measures could be added.

5.4 The Client

In the prototype, a basic client implementing the Jabber protocol was used to send and receive messages to and from the agents.

5.4.1 The Protocol

The protocol used in the dialogue with the agents is very interesting from a usability point of view. With intelligent agents comes the possibility of bringing the syntax of the communication rather close to a natural language dialogue. However, when the user is typing messages into a regular instant messaging client, a tradeoff with the need for brevity has to be made.

In the prototype, only the support for simple X10 devices was implemented, so the protocol was kept equally simple. It allows for commands like “Turn on the heater”, queries like “Is the heater on?” and subscriptions like “Tell me when the door is open”.

5.5 Security

Apart from the server requiring clients and agents to log in, only the basic security measure of rejecting messages from unknown users has been taken in the prototype. This feature should be moved to the server and be replaced by some proper form of message authentication, as described above in *Remote Controlling Issues*.

5.6 Configuration and Setup

An interface for configuration and setup of the system is not included in the prototype. However, it must be recognised that an easy configuration and setup would be essential for the usability of this system. Especially since it is not conceptually trivial to use one single method of control for devices and services which might have very diverse natures and interfaces.

The interface for configuration and setup would most likely be web based, running for example on the Yaws server. This would allow for remote configuration, and since the Yaws server is written in Erlang, it could be directly interfaced with the rest of the system. The web interface should be carefully designed with high usability in mind.

6. VALIDATING THE IDEAS

To get an up-to-date, market oriented view on home automation technology, and the industry’s view on the ideas of remote controlling using instant messaging, a visit was made to the Smart Home Show [12]. A number of semi-structured interviews was conducted with ten exhibitors (out of the about 60 companies that the exhibition featured). The selection was aimed at companies with a general product range (as opposed to specialised in e.g. the entertainment segment), and with a good position in the market. The purpose of these interviews was to get the view of manufacturers and suppliers on:

1. The current market situation: drivers and obstacles in the adaptation of home automation technology
2. Remote controlling of home automation devices: what exists today?
3. Potentials of using instant messaging for such remote controlling

6.1 Results of Interviews

Below is a summary of the outcome of the interviews described above.

6.1.1 The Home Automation Market

- Simple, low-end products such as X10 create interest for the home automation concept.
- The entertainment segment of home automation is one of the main market drivers.
- The biggest obstacle for large-scale adoption of home automation technology is cost.

6.1.2 Remote Controlling

- There is an interest among manufacturers of home automation products in the possibility of remote controlling devices in the automated home.

- Some possibilities already exist: using a web interface and using touch tones from a telephone.
- The most likely way of interfacing with devices in the future is IP[3]. We will see more and more IP enabled devices.
- Instant messaging would be suitable for all sorts of monitoring purposes.
- Security is a main issue in remote controlling.

The Erlang Questions mailing list [11] was another useful resource for getting feedback from the Erlang developer community on the ideas, and gaining knowledge about related or similar ideas recently developed or under development.

7. DISCUSSION

What are today the main issues in the remote controlling of devices in a home automation context?

A general problem on the home automation market is that it is still, at least in Europe, in its "early adoption" phase. Related to this are issues of limited interoperability, limited supply of services and products, and also high costs. On the technical side, security and scalability are two main issues. It is reasonable to assume that the market for home automation will increase substantially and grow more sophisticated. This applies especially to Europe, considering the size and growth of the US market.

What are the benefits of using instant messaging technology for the remote controlling of devices and services?

The concept of instant messaging fits well with the types of communication involved in device remote controlling, including device feedback and status. It is a good way of representing diverse forms of communication with diverse forms of devices.

The server based architecture of instant messaging provides location independence and scalability and makes it possible to offer remote access facilities as a service to a large number of users.

Using the existing servers, clients, protocols, concepts and user interfaces of instant messaging gives a high level of reuse and ease of adoption for users. Clients can be run on a broad range of devices, including PDAs, mobile phones and lap tops.

How could an intelligent instant messaging based solution for remote controlling be implemented?

We have suggested a design for an intelligent gateway based on instant messaging and the concept of agents. We have also shown how this design can be implemented using Erlang/OTP. The transparent multi-protocol nature of the gateway addresses the issue of interoperability from a usability point of view, since it enables a single point of access to many different devices and services. In this sense it could be seen as a step towards the "converged client".

The Converged Client

The way of applying instant messaging described in this paper is a means of providing a single, location independent gateway to a set of devices (or services). This could be seen as a step towards something described as the "Convergence of Clients of Communication Systems" by Olatunde and Grimm [5]. This refers to the convergence of interfaces to devices/services "…such that a client device (converged client) that implements the converged interface is able to access the available services in any backend device dynamically in an ad hoc fashion, and thereby replace the many client devices that would have been used normally".

[3] An IP (Internet Protocol) enabled device can connect directly to the Internet (or a LAN).

7.1 Future Work

The main limitation to what the system could allow is the client, both regarding functionality and security. It must be noted, however, that modifying the client takes away the reuse benefit of using existing clients. Below, some of the limitations of common clients today are discussed from a functionality and security point of view.

Functionality

The notion of status could be used in more sophisticated ways if it were not for the fact that clients generally are limited to a predefined set of often irrelevant status settings, such as "do not disturb" and "invisible".

It would be desirable to add voice recognition as a way for the user to communicate with the agents. This would not necessarily require any modifications of the client. Instead, the server could be equipped with a voice recognition interface which receives ordinary phone calls and produces plain text messages which are then passed on to an agent. Such a voice remote control could possibly be of great interest from an accessibility point of view.

Security

Messages need to be uniquely identifiable for the system to be protected against replay attacks. We were not able to find a client with this level of security, although it is not unlikely that one exists or is under development.

Many clients do not support encryption either. This is however a minor problem, since it is not uncommon for clients to support SSL.

7.2 Related Work

Some solutions based on related ideas have been located. Two of them are described below.

- Werthschulte and Schneider [4] describe how access to an EIB installation through a web browser was created.
- A suggestion on how extensions of the SIP protocol could be used for remote controlling is made by Roychowdhury and Moyer [3]. They also outline a gateway style solution for remote access to networked appliances.

SIP (Session Initiated Protocol)

SIP [16], which like Jabber is an XML based protocol with presence features, has been suggested for purposes similar to the ones described in this paper and an instant messaging extension has been defined for it (Roychowdhury and Moyer [3]). It might therefore be an alternative to Jabber. It is however outside of the scope of this paper to compare the protocols' suitability. Also note that work has been done on the interoperability of these two protocols.

8. CONCLUSION

In this paper, we have demonstrated that it is possible to remote control devices from an instant messaging client, and that instant messaging enables high scalability, usability and reuse. We have described an original concept for remote controlling, shown how this can be implemented and how intelligence can be added to the system.

We have also demonstrated how the reuse of open source systems and Erlang/OTP made it possible to rapidly create an environment so that we could focus on the functionality of the system and understanding the higher level issues.

9. ACKNOWLEDGEMENTS

University supervisor: Annika Aasa, IT University of Gothenburg.

Industry supervisors: Marcus Taylor and Francesco Cesarini, Erlang Training and Consulting Ltd.

Mickaël Rémond, author of the Jabberlang library, which was used for the implementation of XMPP connections.

Daniel B Suthers, author of the Heyu X10 controller, used in the prototype as an X10 device driver.

10. REFERENCES

[1] K. Werthschulte, G. Westermeir, F. Schneider. "Secured Data Transmission for Control and Supervision of an EIB Installation Using Mixed Network Topologies", Technische Universität München/EIB Event, 2001.

[2] *The X10 Protocol:* http://www.x10.com/technology1.htm Technica Pacifica. "Easy X10 Projects For Creating A Smart Home", Indy-Tech Publishing, 2005.

[3] A. Roychowdhury, A. Moyer. "Instant Messaging and Presence for SIP Enabled Networked Appliances", (Publisher unknown), 2001. http://www.voip.nce.ufrj.br/courses/graduacao/lc4/Instant Messaging and Presence for SIP Enabled Networked Appliances.pdf

[4] K. Werthschulte, F. Schneider, "Providing Access to an EIB-Installation using Internet Browser Technology", Technische Universität München/EIB Event, 2000.

[5] O. P. Olatunde, A. Grimm. "Convergence of Clients of Communications Systems", International Wireless Congress: Systems & Applications ZVEI Elektronika. ICM Munich. November 2004.

[6] Erlang/OTP (Open Telecom Platform) http://www.erlang.org

[7] Supervision Principles in Erlang/OTP http://www.erlang.se/doc/doc-5.4.3/doc/design_principles/sup_princ.html#5.1

[8] XMPP, Extensible Messaging and Presence Protocol Defined in the *IETF* specifications RFC 3920 (http://www.ietf.org/rfc/rfc3920.txt) and RFC 3921 (http://www.ietf.org/rfc/rfc3921.txt)

[9] MSN Messenger http://messenger.msn.com

[10] Skype http://www.skype.com

[11] Erlang Questions mailing list http://www.erlang.org/ml-archive/erlang-questions/

[12] The Smart Home Show http://www.smarthomeshow.co.uk/

[13] Jabber instant messaging technology http://www.jabber.org/about/overview.shtml

[14] Yaws, open source web server written in Erlang http://yaws.hyber.org/

[15] Ejabberd – open source instant messaging server http://ejabberd.jabber.ru/

[16] SIP – Session Initiation Protocol http://www.cs.columbia.edu/sip/