The HomeSIP Project: Using Open Source SIP Stack for Home Automation

P. Kadionik, A. Ben Atitallah, P. Nouel
IMS, ENSEIRB, University of Bordeaux, France

email : kadionik@enseirb.fr
web : http://www.enseirb.fr/~kadionik
http://www.enseirb.fr/cosynux/

C. Consel, L. Burgy, N. Palix, W. Jouve et J. Lancia
LaBRI, ENSEIRB, University of Bordeaux, France

INTRODUCTION

The HomeSIP Project

- The HomeSIP project (acronym for *Home Automation with SIP*) consists to setup a HW/SW platform based on the SIP protocol.
- This project is oriented embedded systems for Home Automation and is built with different hardware devices running free software (embedded Linux, protocol stacks…).
- It will be used at the ENSEIRB School of Electrical Engineering (*Ecole Nationale Supérieure d'Informatique Electronique et Radiocommunications de Bordeaux*)

The HomeSIP Team

HW/ Firm

- Cosynux Team from the IMS Lab (Intégration du Matériau au Système):
  - COnception de
  - SYstèmes
  - NUmériques X

(with the Linux Touch!)

SW/ DSL

- Phoenix Team from the LaBRI Lab (Laboratoire Bordelais de Recherche en Informatique).

- More informations:
  http://www.enseirb.fr/cosynux/
  http://phoenix.labri.fr/

THE HOMESIP PROJECT AND M2M

The Idea

- The HomeSIP project is based on the well known SIP protocol (*Session Initiation Protocol*) and generally used for VoIP (*Voice Over IP*).
- The idea is to used SIP as an universal container for collecting data coming from sensors (inputs) and for driving actuators (outputs).
- SIP is also used in a Home Automation context!

The HomeSIP Project Overview

- The HomeSIP project has different components:
  - An hardware platform composed with sensors and actuators connected to embedded systems with network capabilities and having an Internet connectivity.
  - Different specific software running Linux in the embedded systems.
  - A new dedicated DSL language (*Domain Specific Language*) pour developing new services on the HomeSIP platform.
- The HomeSIP project will be finally integrated to the VoIP platform of the ENSEIRB School of Electrical Engineering for real testing…

HomeSIP and M2M

- The HomeSIP project is also a project for Home Automation that is a part of the M2M engineering.
- M2M (*Machine to Machine*) is an infrastructure based on an network that allows communications directly between devices or through a server without human control. It allows for example automatic data recording and processing by the devices.
- It is in fact the logical evolution of remote control of an HW device connected to the Internet!

M2M

- M2M is the convergence between electronics, software computing and networking. The Internet network is used as a backbone.

- New concepts have emerged like:
  - Ubiquitous Internet.
  - Ubiquitous computing.
  - Pervasive computing.

- Wireless networks (with IP protocol) are generally used:
  - Bluetooth, Wifi, ZigBee…
  - Example: wireless sensor networks.

M2M

- This new class of applications corresponds to activities consisting in collecting, coordinating, transmitting, treating and reacting to information emanating from communication entities (or objects), local or distant, fixed or mobile.

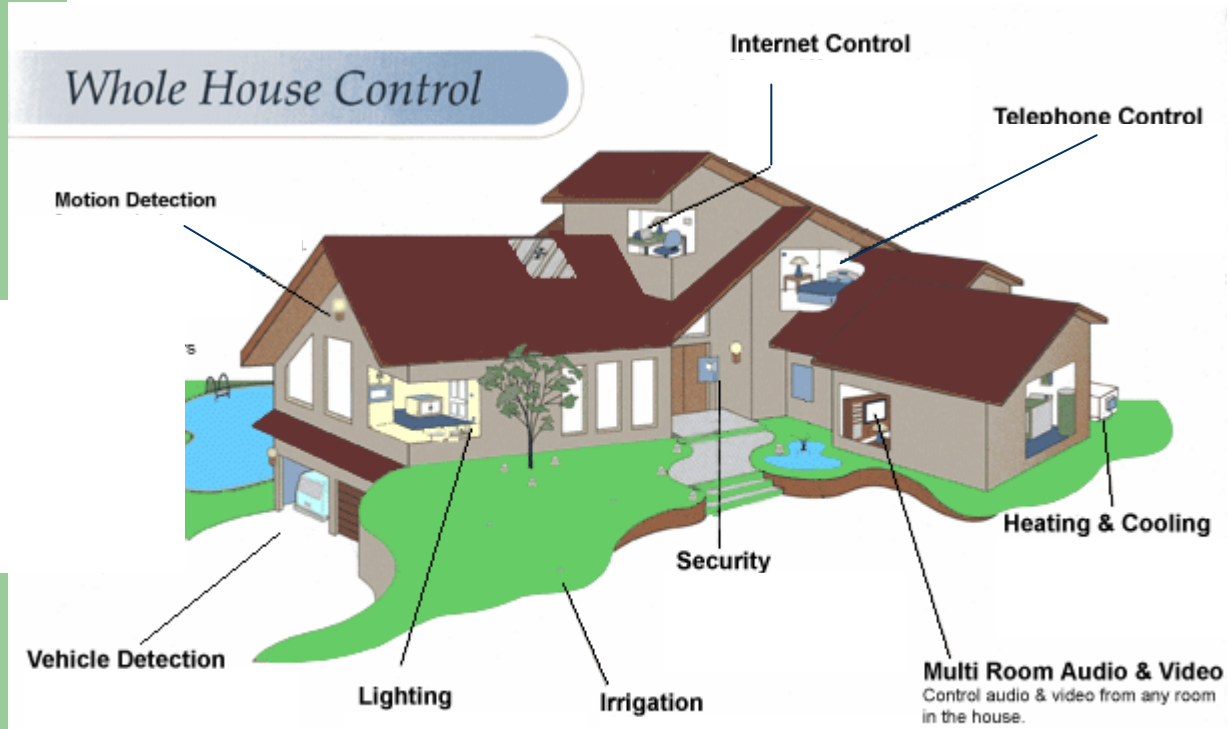
- The communication infrastructure manages all the aspects of the communication between the entities of an M2M application (abstraction from HW).

M2M

- M2M has several applications like:
  - Home Automation, building management.
  - Security.
  - Video monitoring.
  - Navigation
  - Telemetry.
  - Energy monitoring.
  - Remote medicine.
  - Electronic money.
  - …

IP PROTOCOLS FOR HOME AUTOMATION

The Home Automation Context



- Receive Alarm
- Receive regular data
- Global control interface
- Multi-users
- Remote control

An Home Automation example

The Home Automation Context

- From the Home Automation point of view, we can distinguish two kinds of data transfers:
  - **Synchronous transfers**. The current data is sampled by an application, for example the current value from a sensor. We want to perform an action too, for example, changing the current value of an actuator. A sensor may be a temperature sensor and a actuator, a simple electric bulb.
    - We must perform GET and PUT actions.
  - **Asynchronous transfers.** An alarm is sent asynchronously to the user when something is going wrong. For example a sensor is detecting a malfunction.
    - We must collect ALARM events.

Internet Protocols for Home Automation

- In the constellation of Internet protocols, we must choose those supporting GET, PUT, ALARM actions for controlling hardware devices.

- The most interesting protocols are:
  - HTTP.
  - SMTP.
  - SNMP.
  - SIP.
  - Special case: Specific network application

Specific Network Application for Home Automation

- The specific network application is:
  - Based on the socket API.
  - The data stream is byte oriented and not structured.
  - Dedicated and specific.
  - UDP: connexionless and unreliable.
  - TCP: connection oriented and reliable.
  - Quite difficult to develop.
- For Home Automation, a specific network application may support GET, PUT and ALARM actions.

HTTP for Home Automation

- The HTTP protocol is:
  - The acronym for HyperText Transfer Protocol.
  - A RFC.
  - A client/server application. The client is a web browser and the server is the web server embedded into the HW device.
  - The data stream is ASCII character oriented. HTTP has ASCII commands and ASCII responses.
  - Uses TCP transport protocol.
  - Quite easy to use.
- For Home Automation, HTTP supports GET and PUT actions.

SMTP for Home Automation

- The SMTP protocol is:
  - The acronym for Simple Mail Transfer Protocol.
  - A RFC.
  - A client/server application. The client is the mail client embedded into the HW device and the server is the mail server.
  - The data stream is ASCII character oriented. SMTP has ASCII commands and ASCII responses.
  - Uses TCP transport protocol.
  - Quite easy to use.
- For Home Automation, SMTP supports ALARM actions.

SNMP for Home Automation

- The SNMP protocol is:
  - The acronym for Simple Network Management Protocol.
  - A RFC.
  - A client/server application. The client is a SNMP manager and the server is the SNMP agent embedded into the HW device.
  - The data stream is byte oriented and coded according the BER Rule (Basic Encoding Rule). SNMP has commands and responses.
  - Uses UDP transport protocol.
  - Quite difficult to use and to understand.
- For Home Automation, SNMP supports GET, PUT and ALARM actions.

SIP for Home Automation

- The SIP protocol is:
  - The acronym for Session Initiation Protocol.
  - A RFC.
  - A client/server application. The client is a SIP Client User Agent and the server is the SIP Server User Agent embedded into the HW device.
  - The data stream is ASCII character oriented. SIP has ASCII commands and ASCII responses.
  - Uses UDP transport protocol. TCP is possible.
  - Quite easy to use.
- For Home Automation, SIP supports GET, PUT and ALARM actions.

Resume

Protocol	Transport Mode	Stream	Get	Put	Alarm	Easy to use?
Specific network application	TCP UDP	byte ***(ASCII)***	yes	yes	yes	no
HTTP	TCP	ASCII	yes	Yes ***(through CGI)***	No	yes
SMTP	TCP	ASCII	-	- ***(embedded mail server?)***	yes	yes
SNMP	UDP	byte	yes	yes	yes	no
SIP	UDP ***(TCP…)***	ASCII	yes	yes	yes	yes

SIP versus SNMP

- The SNMP protocol has the same capabilities than SIP for Home Automation:
  - The SNMP GET message.
  - The SNMP PUT message.
  - The SNMP TRAP message for alarms.

- But:

SIP versus SNMP

- The SNMP protocol is not well suited for Home Automation:
  - The SNMP protocol is complex (for example with the BER encoding).
  - It is complex to use. The SNMP agent extension is something complicated (see NET-SNMP).
  - The SNMP protocol is unsecure (with the most used version: 1).
  - The SNMP agent has not a small memory footprint and is not well adapted for tiny embedded systems.
  - These are no popular client applications. There's a lot with SIP!
  - ...
- **SIP is also a better choice!**

SIP PROTOCOL PRESENTATION

General SIP Structure

- The SIP protocol (*Session Initiation Protocol)* is a package of different RFCs:
  - RFC 3261: *SIP: Session Initiation Protocol*.
  - RFC 3265: *Session Initiation Protocol (SIP)-Specific Event Notification*.
  - RFC 3428: *Session Initiation Protocol (SIP) Extension for Instant Messaging*.
- SIP is a signaling protocol for setting a session between 2 users (or devices) for any kind of data exchange (or stream) via Internet.

General SIP Structure

- SIP comes from more classical signaling protocols like CCS7 (Common Channel Signaling 7) protocol from the ITU.

- CCS7 is used in the POTS (Plain Old Telephone Service), ISDN and GSM networks.

General SIP Structure

- But SIP is more flexible and is not dependent to any kind of data stream nor to any kind of transport network!

- This quality is its strength. SIP is used for putting two users into relation for exchanging any kind of data:
  - VoIP. The well known case!
  - Instant messaging.
  - Video.
  - Others : presence indication, event notification…

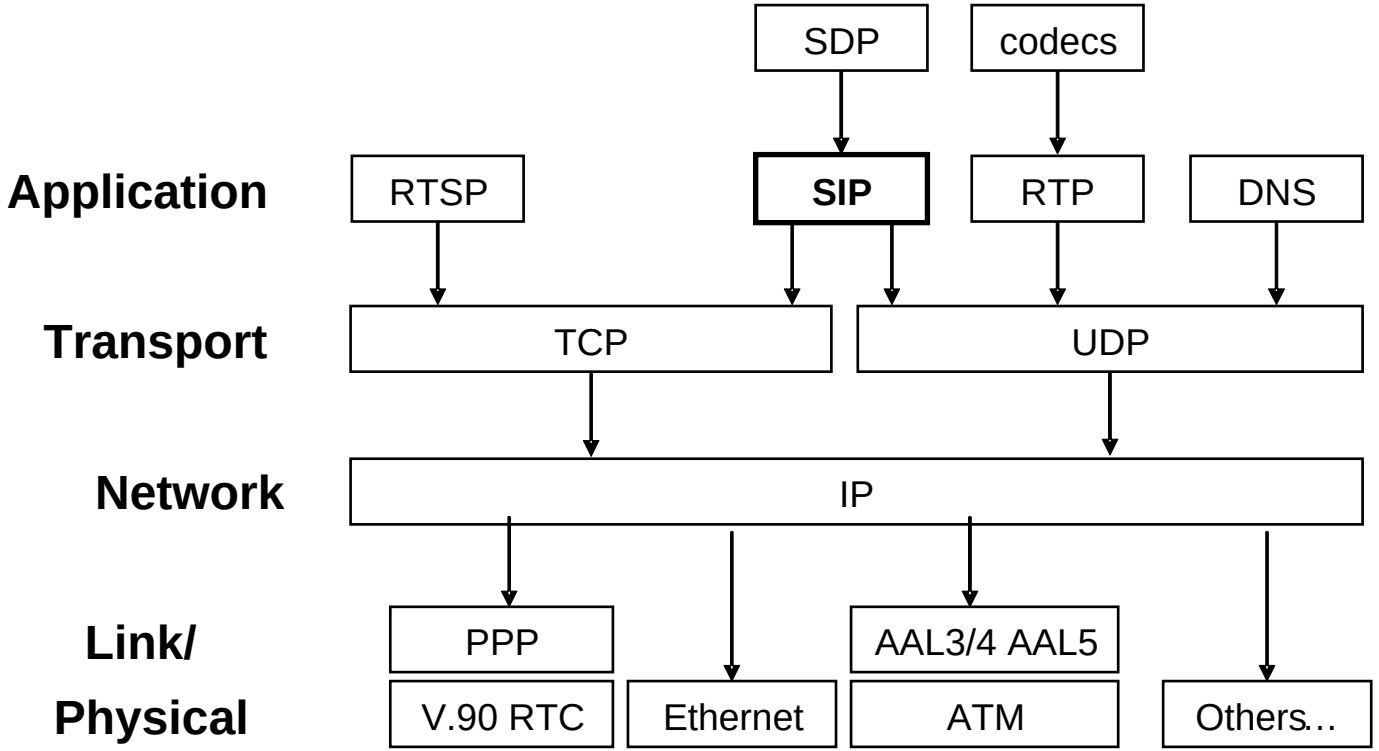
General SIP Structure

- SIP messages can be transported over:
  - The UDP protocol. The most used case.
  - The TCP protocol.

And:

  - The TLS protocol (*Transport Layer Security*, RFC 3546) using SSL (*Secure Socket Layer*) over TCP.
  - The SCTP protocol (*Stream Control Transport Protocol*, RFC 2960).

General SIP Structure



SIP protocol in the Internet protocol constellation

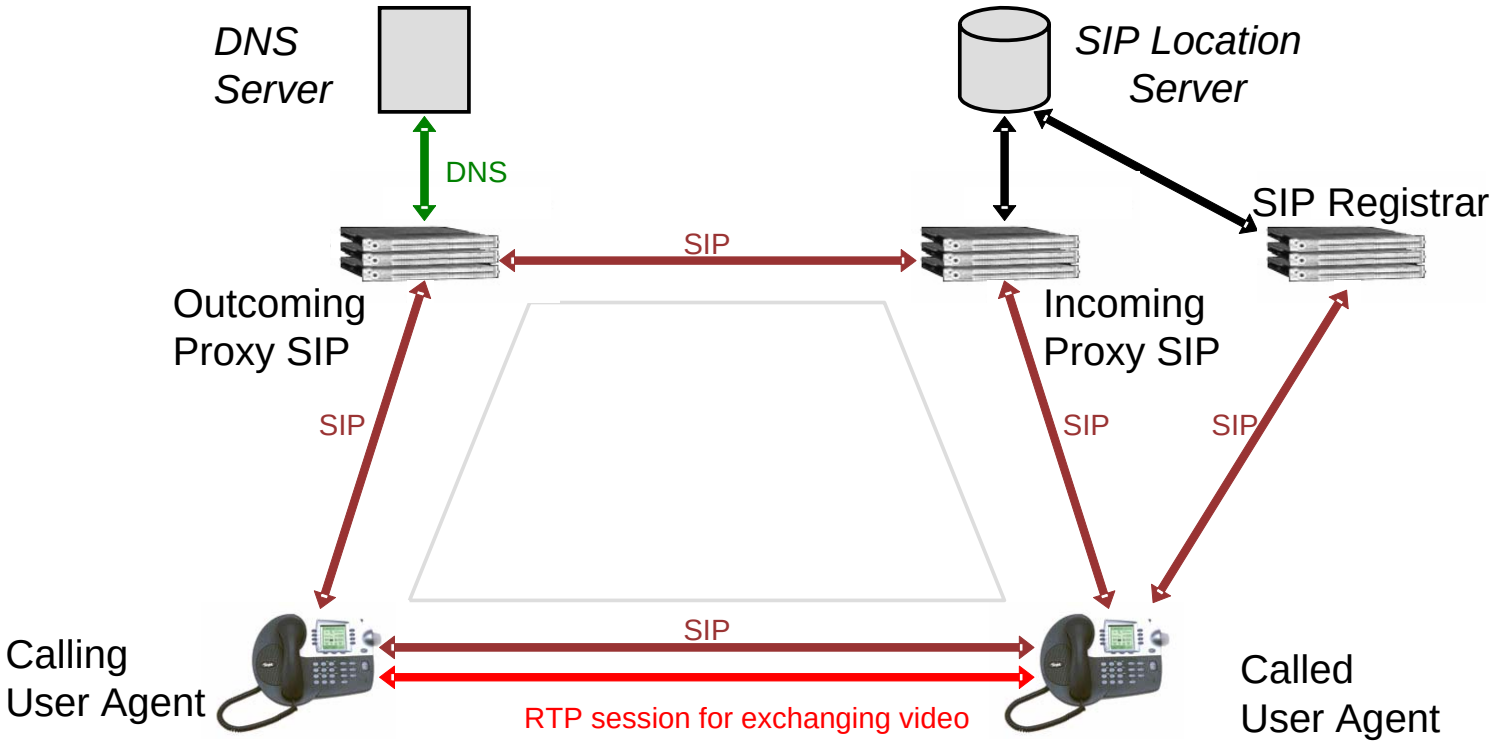
General SIP Structure

- The SIP protocol uses the same concepts than SMTP and HTTP protocols:
  - SIP is based on the client/server programming. Well known SIP port: 5060.
  - SIP dialog protocol is command/response oriented. SIP has ASCII commands and ASCII responses.
  - SIP has an addressing scheme for identifying SIP users (entities): URI (*Uniform Resource Identifier*) like the HTTP URL (*Uniform Resource Locator*).
  - URI example: sip:kadionik@enseirb.fr

SIP Infrastructure

- Several entities (equipments and/or applications) are used by SIP :
  - The SIP User Agent. The Client UA (C-UA) starts the SIP call and emits a SIP request and the Server UA (S-UA) responds to the SIP request.
  - The SIP Redirect Server helps to locate a SIP User Agent.
  - The SIP Registrar Server records the availability of a SIP User Agent. It's processing the SIP REGISTER message. It can be integrated into a SIP Proxy Server.
  - The SIP Proxy Server. It acts as an HTTP proxy.
  - The SIP Location Server. It is a database used by the Redirect Server and/or the Proxy Server.

SIP Infrastructure

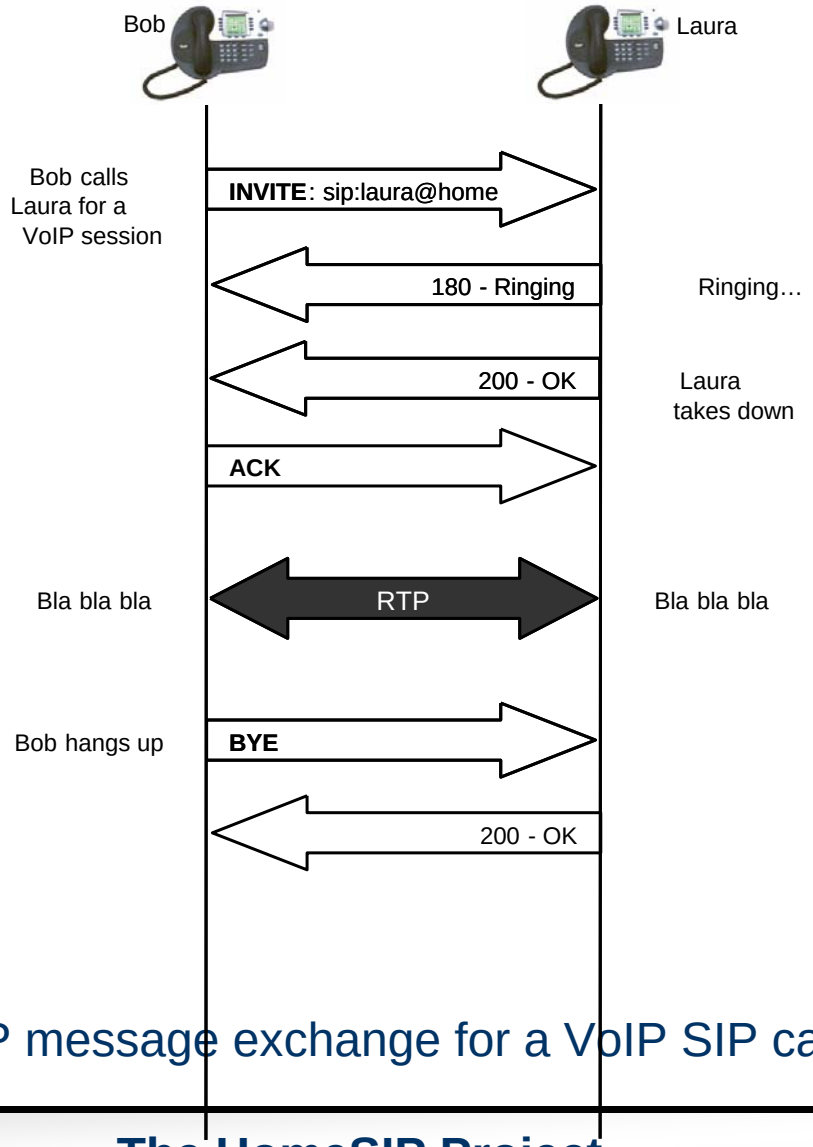


SIP Infrastructure

SIP Messages

- The SIP messages are:
  - REGISTER: for recording a SIP User Agent to the Registrar Server.
  - INVITE: for setting a session between 2 SIP User Agents.
  - CANCEL: for cancelling an INVITE request.
  - ACK: for acknowledging an INVITE request.
  - BYE: for closing a SIP session.
  - SUBSCRIBE, NOTIFY: SIP event subscription and notification.
  - MESSAGE: instant messaging.
  - Others: REFER, OPTIONS, INFO.

SIP Session



SIP message exchange for a VoIP SIP call

SIP Message Examples

- SIP INVITE request:

INVITE sip:laura@home.com SIP/2.0
Via: SIP/2.0/UDP pc11.work.com
Max-Forwards: 70
To: Laura <sip:laura@home.com>
From: Bob <sip:bob@work.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:bob@pc11.work.com>
Content-Type: application/sdp
Content-Length: 131

o=Bob 289123451 289123451 IN IP4 111.22.33.44
s=Let us talk for a while
c=IN IP4 111.22.33.44
t=0 0
m=audio 20002 RTP/AVP 0
a=rtpmap:0 PCMU/8000

- SIP Response:

SIP/2.0 200 OK
To: Laura <sip:laura@home.com>;tag=a6c85cf
From: Bob <sip:bob@work.com>;tag=1928301774
Call-ID: a84b4c76e66710
CSeq: 314159 INVITE
Contact: <sip:laura@222.33.44.55>
Content-Type: application/sdp
Content-Length: 131

v=0
o=Laura 289123444 289123444 IN IP4 222.33.44.55
s=Let us talk for a while
c=IN IP4 222.33.44.55
t=0 0
m=audio 41002 RTP/AVP 0
a=rtpmap:0 PCMU/8000

SIP Protocol and Home Automation

- The SIP protocol has several qualities for Home Automation:
  - Supports abstract addressing/naming.
  - Provides security: both authentication/authorization and privacy.
  - Support different communication mechanisms for devices.
  - Provides a flexible payload capability based on MIME types.
  - Inter works with different in-home networking technologies: X.10, CAN bus, CPL. An hardware gateway may be used for this.
  - Provides support for the mobility of networked appliances.
  - Reuses of an existing SIP infrastructure for a whole new set of services.

SIP Protocol and Home Automation

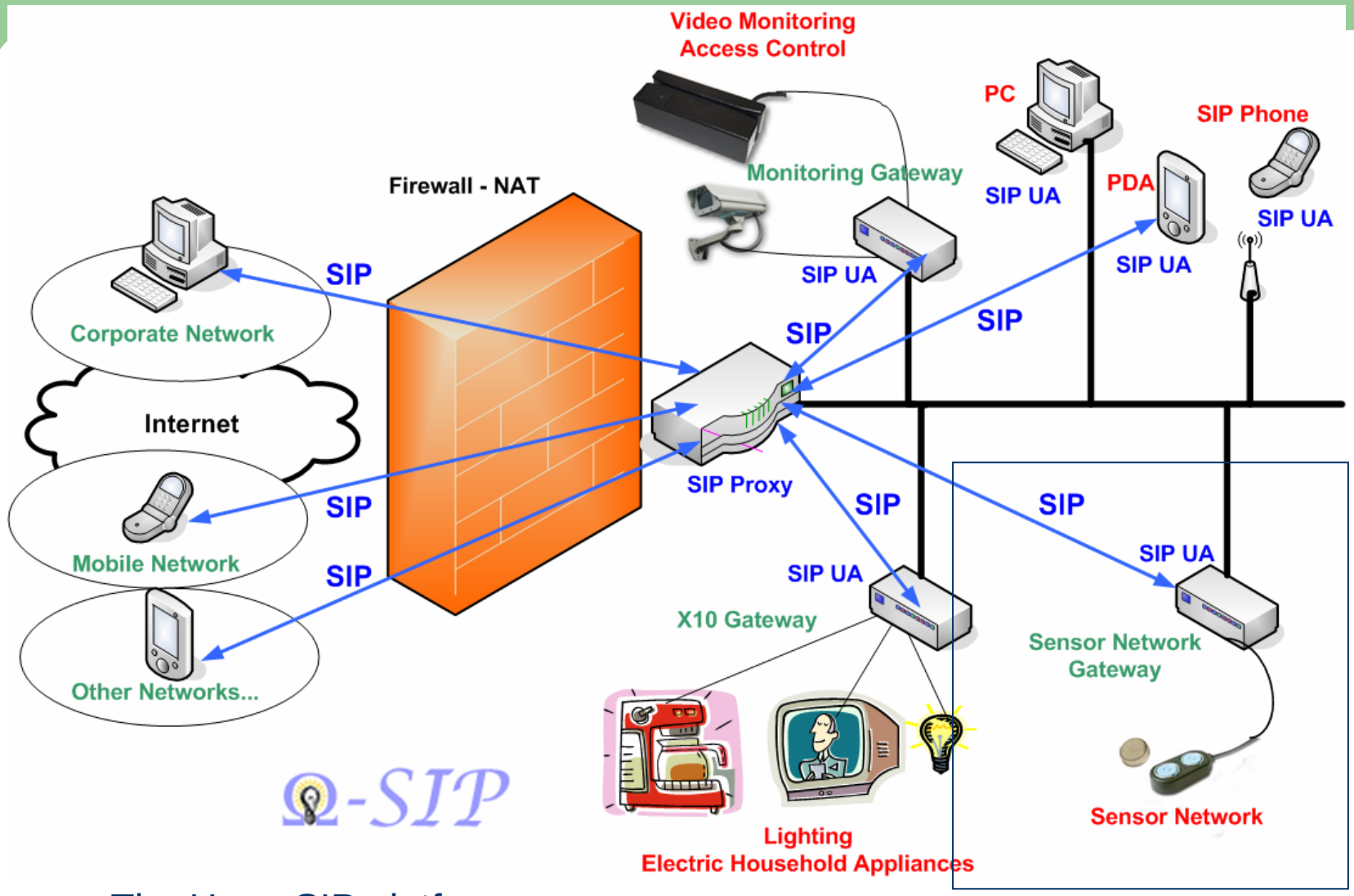
- We can use these SIP messages for Home Automation:
  - SIP MESSAGE message for PUT or GET actions. This message is primary used for instant messaging (RFC 3428).
  - SIP SUBSCRIBE and NOTIFY messages for ALARM events. Theses messages are primary used for presence notification (RFC 3265).

THE HOMESIP PLATFORM: HW and SW DEVELOPMENT

The HomeSIP Platform

- It will be added to the ENSEIRB's VoIP platform.
- SIP devices are PC, SIP phones, dedicated embedded Linux devices (gateways), PDA (Nokia N770).
- At this time, only 2 SIP/sensor network gateways are used during software development.
- It is scheduled to add SIP/X.10 gateways, SIP/video gateways and SIP/RFID gateways...
- The Ethernet network is used as an IP backbone.

The HomeSIP Platform



The HomeSIP platform

HomeSIP: HW development

- The SIP/sensor network gateway is based on commercial hardware in order to avoid developing it:
  - ARM9 SBC board (from Eukréa). This SBC runs embedded Linux and has a lot of I/O for connecting sensors and actuators: serial lines, I2C bus, USB bus, I/O… The SBC board has its own Ethernet interface in order to act as a SIP gateway.
  - Temperature sensors: *iButton*® DS1920 from Dallas Semiconductor with an adapter for serial line connection. The *one Wire* bus used by *iButton*® allows to add other compatible sensors…
  - A specific device has been also developed with a ZigBee interface for wireless connectivity (XBee module with PIC 16F877 with a temperature sensor)

HomeSIP: SW development

- Software development has been done on the SIP/sensor network gateway:
  - Porting Linux on the ARM9 SBC board.
  - Finding an open source SIP stack with low memory footprint and porting it on the SBC board. Perhaps, developing our own?
  - Developing embedded Linux applications for driving sensors.
  - Developing SIP embedded Linux applications for collecting data from sensors through SIP messages.

HomeSIP: Open Source SIP Stacks

- These are several open source SIP stacks we can use. We must choose those written in C or C++ language.
- We have chosen the open source oSIP SIP stack and its extension API eXosip written by Aymeric Moizard.
- The oSIP SIP stack is written in C language, is portable with a very low memory footprint.
- eXosip is an API based on oSIP for simplifying SIP application writing (used in our case).

HomeSIP: Open Source SIP Stacks

- Available open source SIP stacks:

	Sofia-SIP	oSIP	reSIProcate	SipX	Opal	Vocal
VxWorks port	No	Yes	No	No	Yes	No
Win32 port	Soon	Yes	Yes	Yes	Yes	No
Linux port	Yes	Yes	Yes	Yes	Yes	Yes
RFC 3261	Yes	Yes	Yes	Yes	Yes	Partially
RFC 2327 (SDP)	Yes	Yes	Yes	Yes	Yes	Yes
RFC 3264 (O/A)	Yes	Yes	Yes	Yes	Yes	?
RFC 3263 (SIP DNS)	Yes	By the application	Partially	Yes	No	?
RFC 3515 (REFER)	Yes	Yes	Yes	Yes	Yes	Yes
RFC 3262 (100rel)	Yes	No	No	No	No	?
RFC 3311 (UPDATE)	Yes	Yes	No	No	No	?
TCP	Yes	Yes	Yes	Yes	Yes	
UDP	Yes	Yes	Yes	Yes	Yes	
TLS	Yes	Yes	Yes	Not tested	No	
Size	< 500 KB	< 400 KB	< 2,5 MB	< 4 MB	Important	Important
License	LGPL	LGPL	Vovida	LGPL		Vovida

Reference: http://www.huisetalage.nl/sip/stacks.pdf

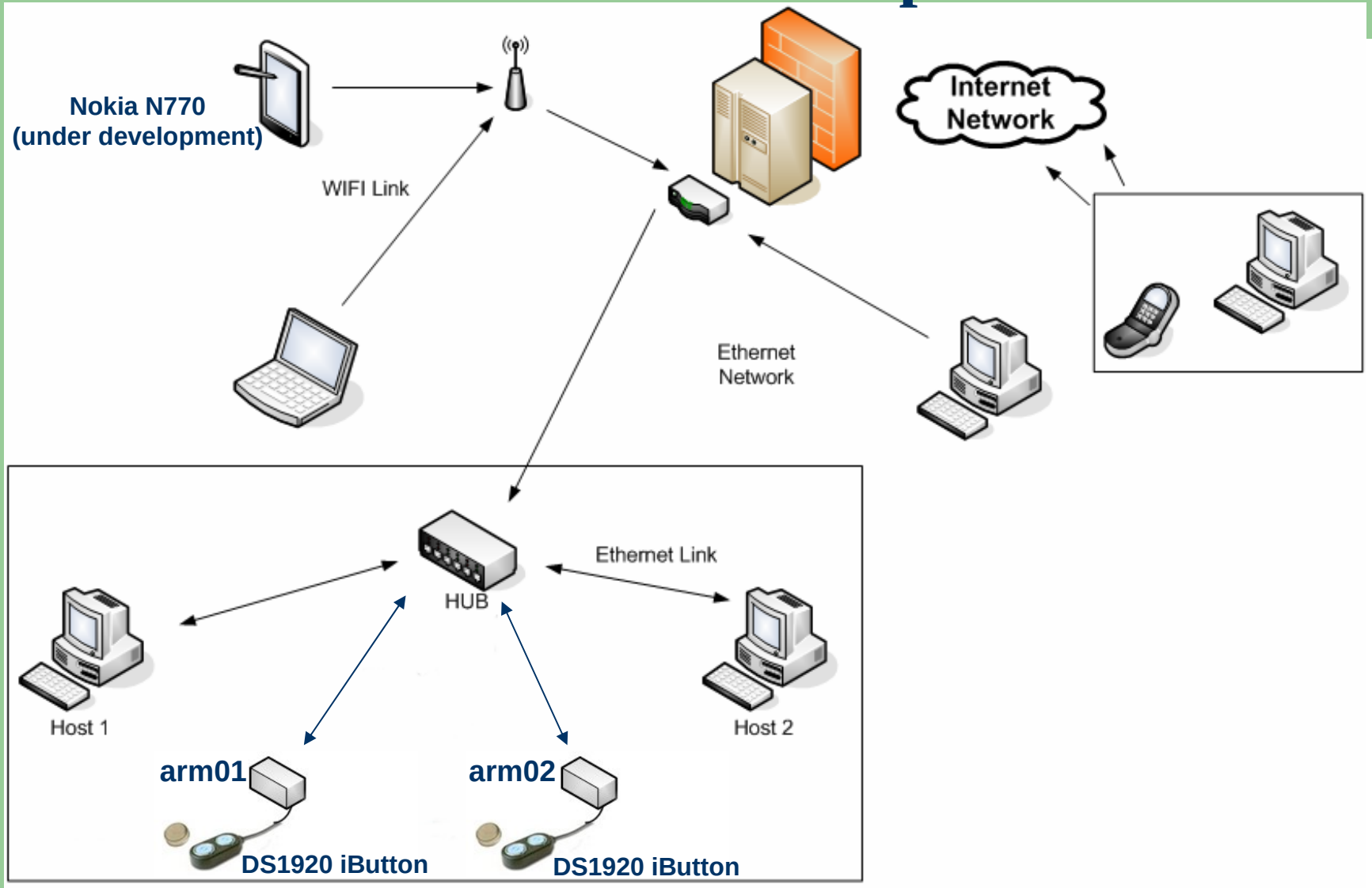
HomeSIP: SW development

- Example 1: a first scenario has been written: periodic sending of the temperature read from a sensor via SIP. The SIP MESSAGE message is used.

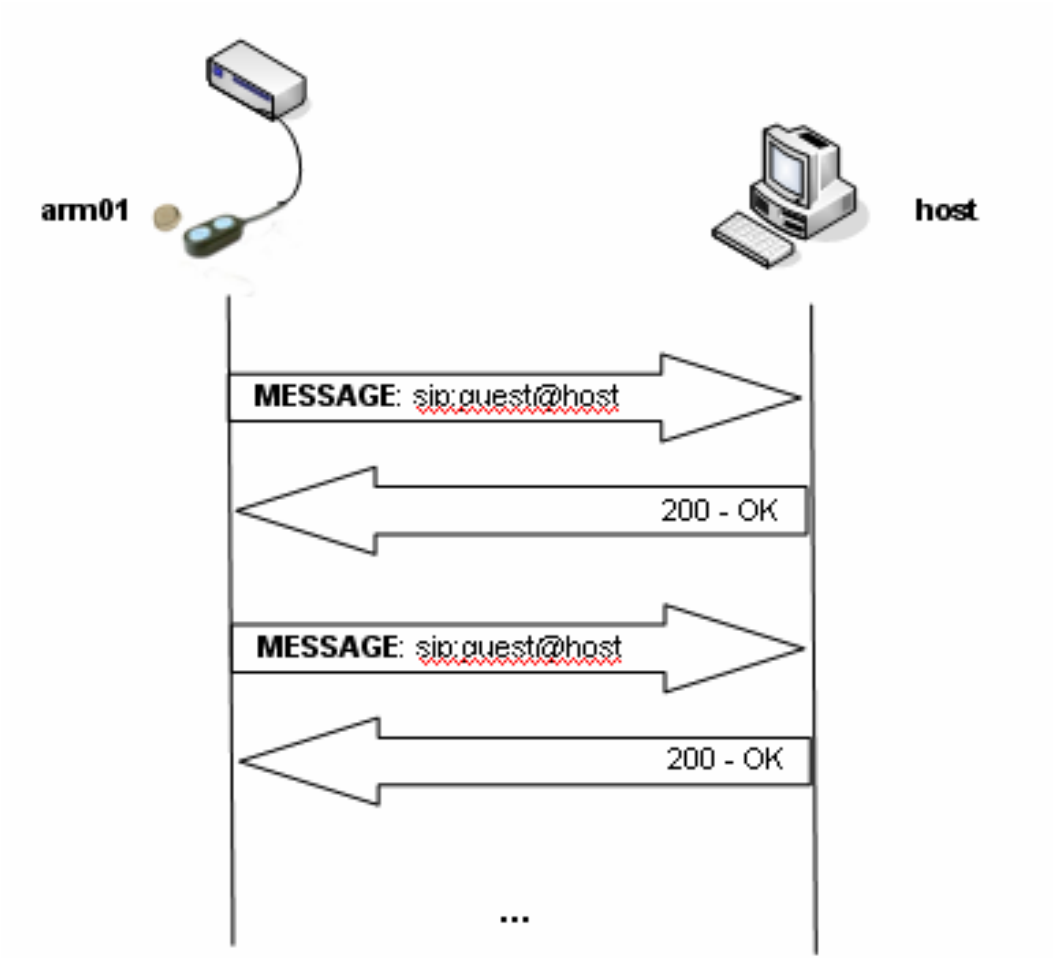
- A SIP User Agent *send_im* embedded in the ARM9 SBC target (arm01) sends an SIP instant message to a SIP User Agent embedded into the host (PC). We use for this *gaim* and the *josua* User Agent (from the oSIP software).

- First case: SIP MESSAGE message for GET actions.

HomeSIP: SW development

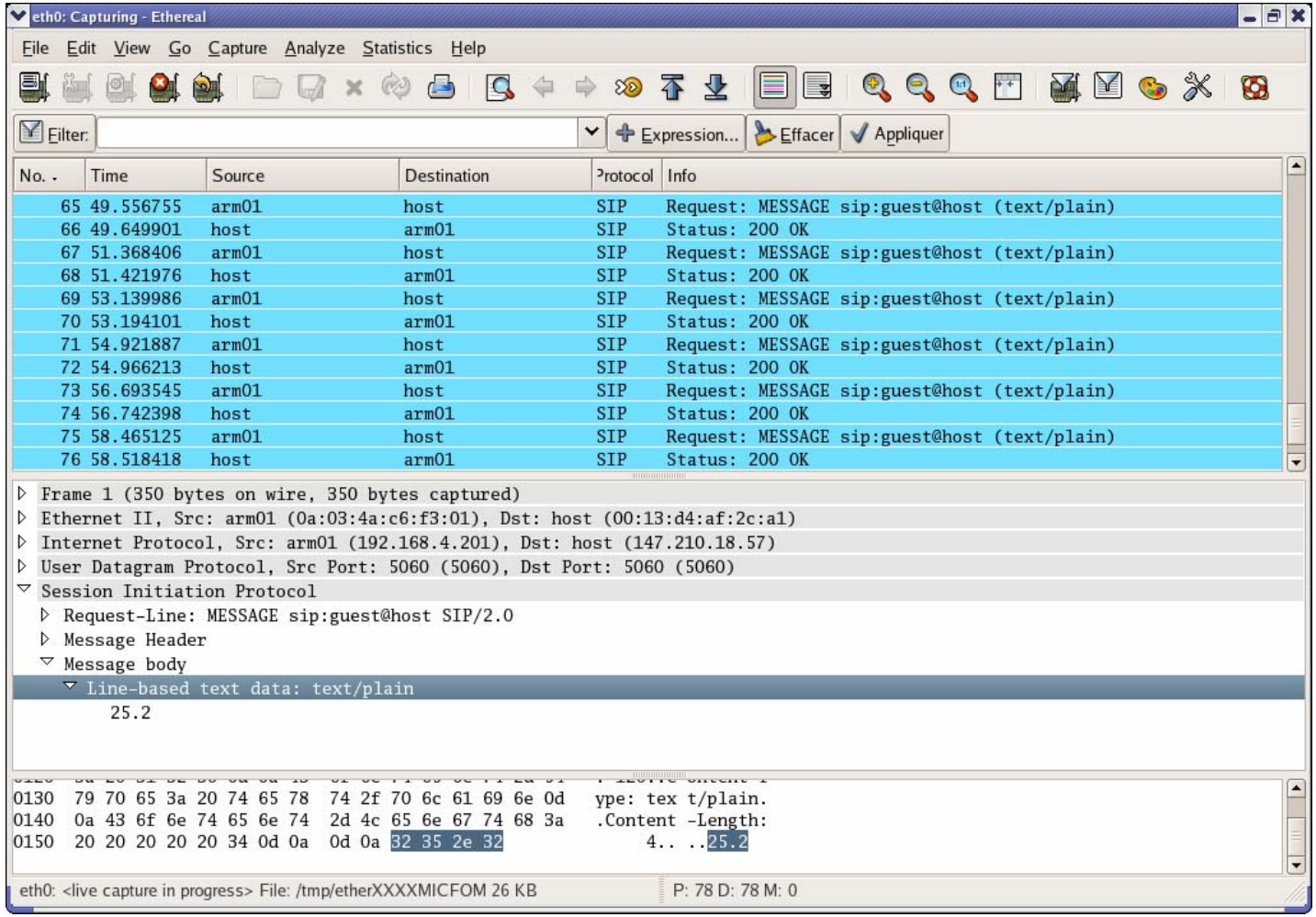


HomeSIP: SW development



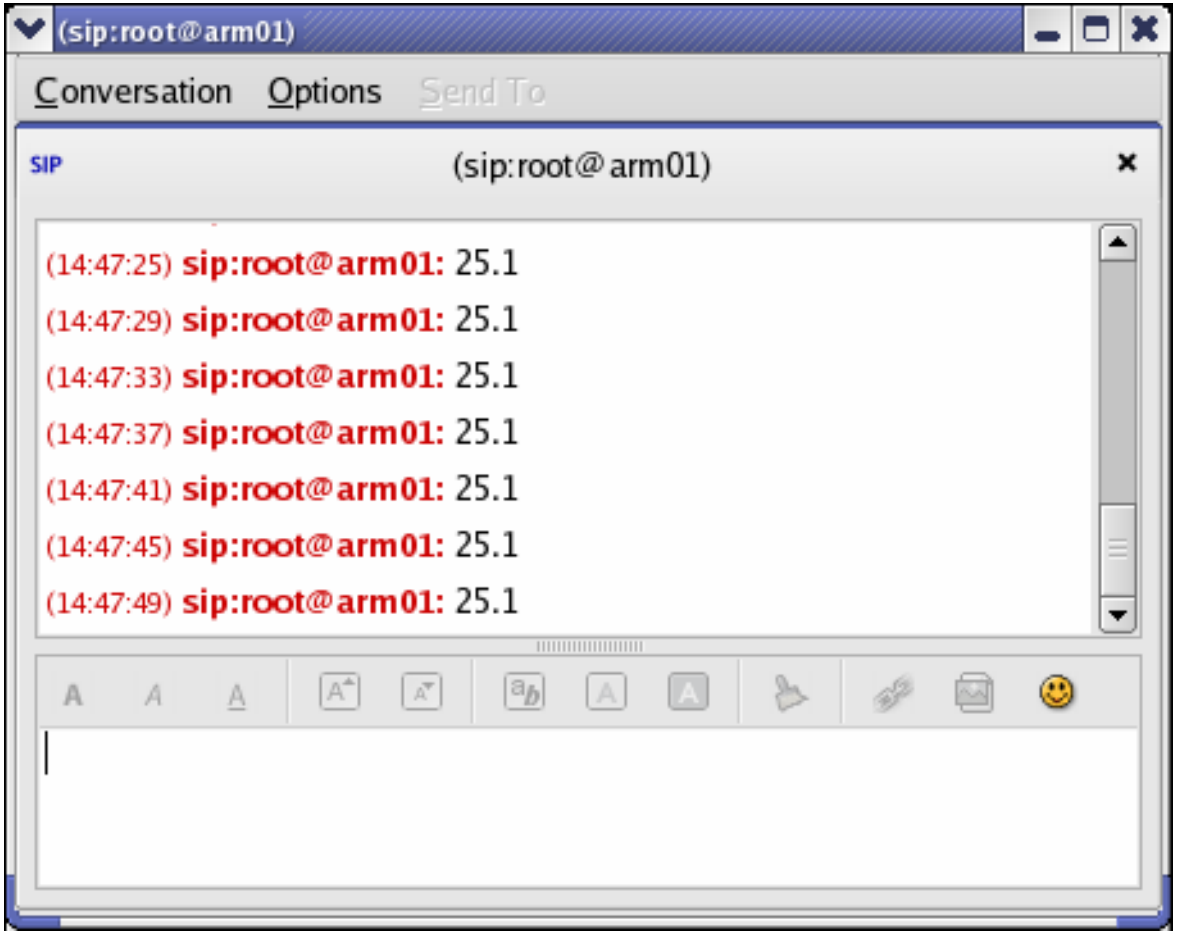
SIP messages generated by send_im

HomeSIP: SW development



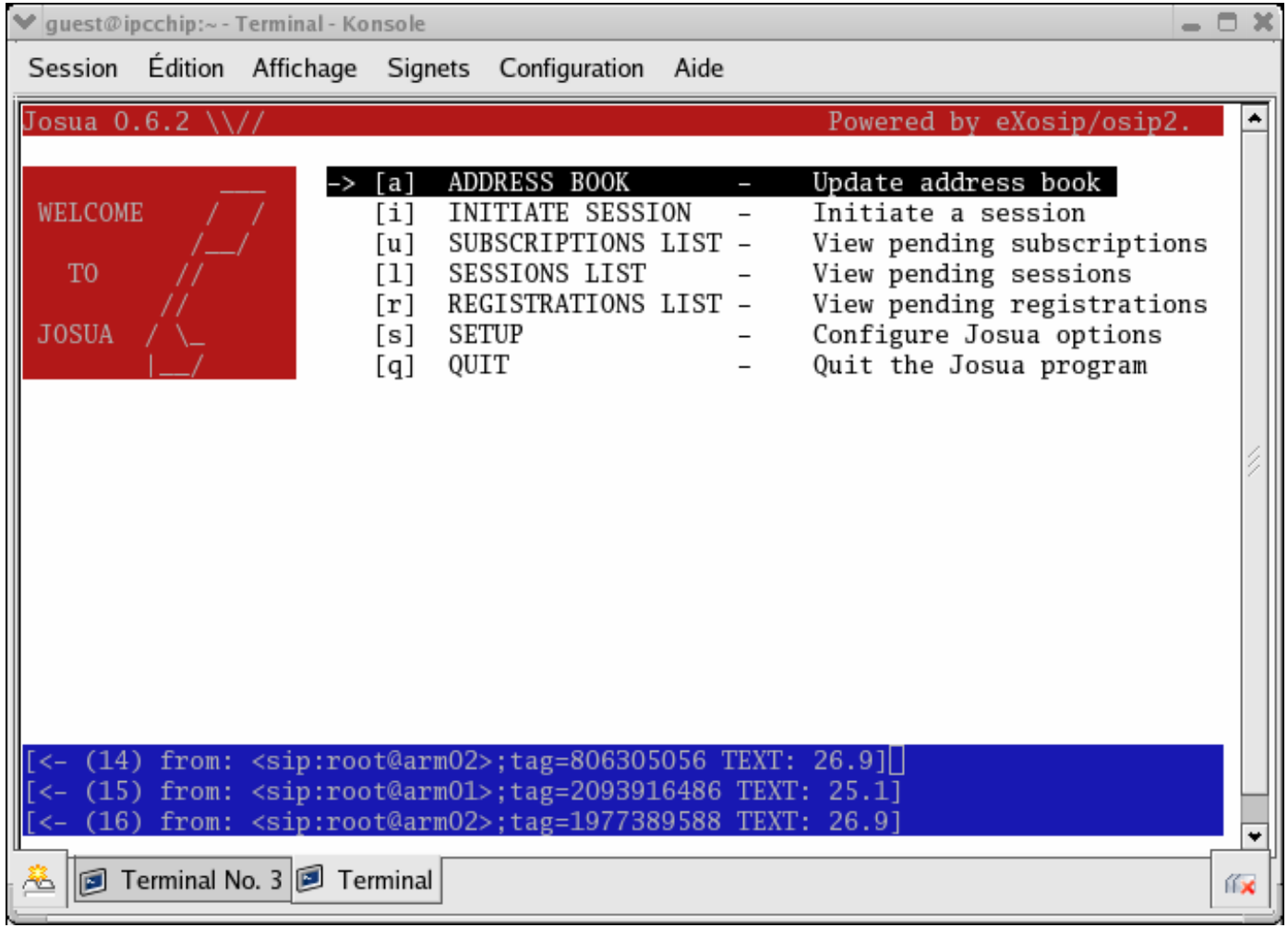
Traces generated by send_im

HomeSIP: SW development



SIP Instant Messages collected by *gaim* (25,1 °C)

HomeSIP: SW development



SIP instant Messages collected by *josua* (25,1 °C et 26,9 °C)

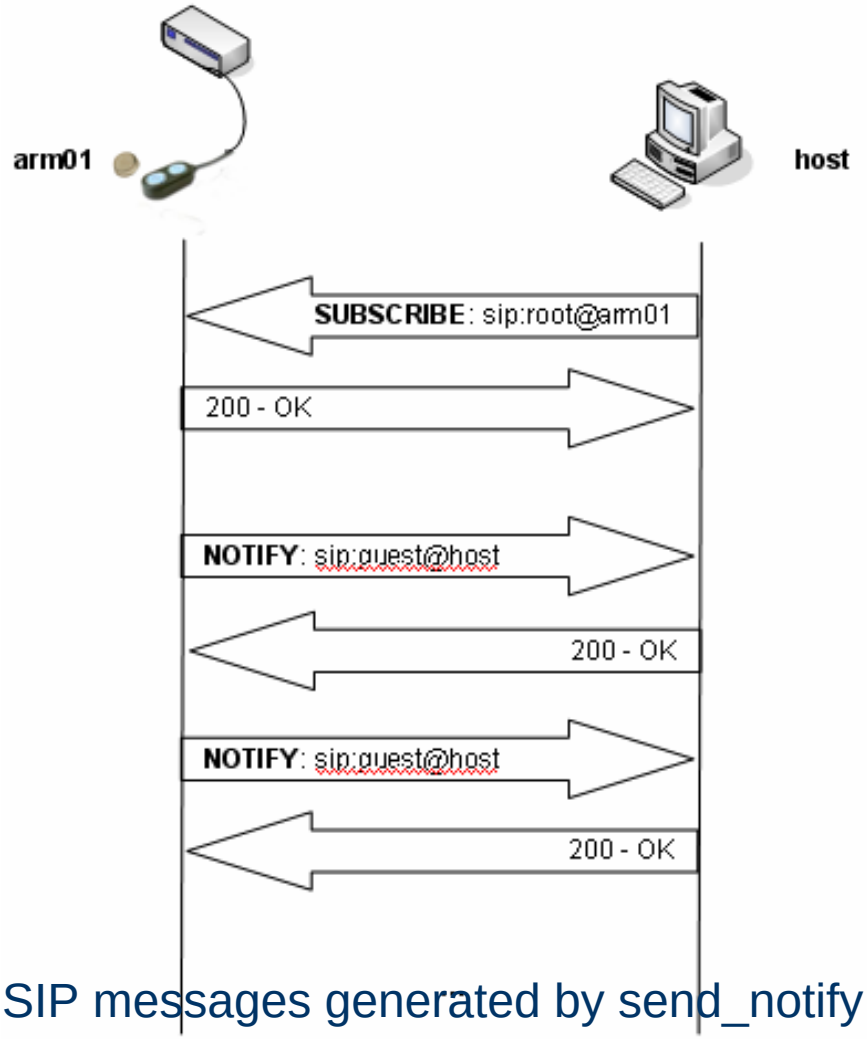
HomeSIP: SW development

- Example 2: A second scenario has been written: asynchronous sending of the temperature read from a sensor via SIP. The SIP SUBSCRIBE/NOTIFY messages are used.

- A SIP User Agent *send_notify* embedded in the ARM9 SBC target reacts to a SUBSCRIBE SIP message and then sends regularly current temperature with a NOTIFY SIP to a SIP User Agent running in the host (PC). We use for this our own User Agent.

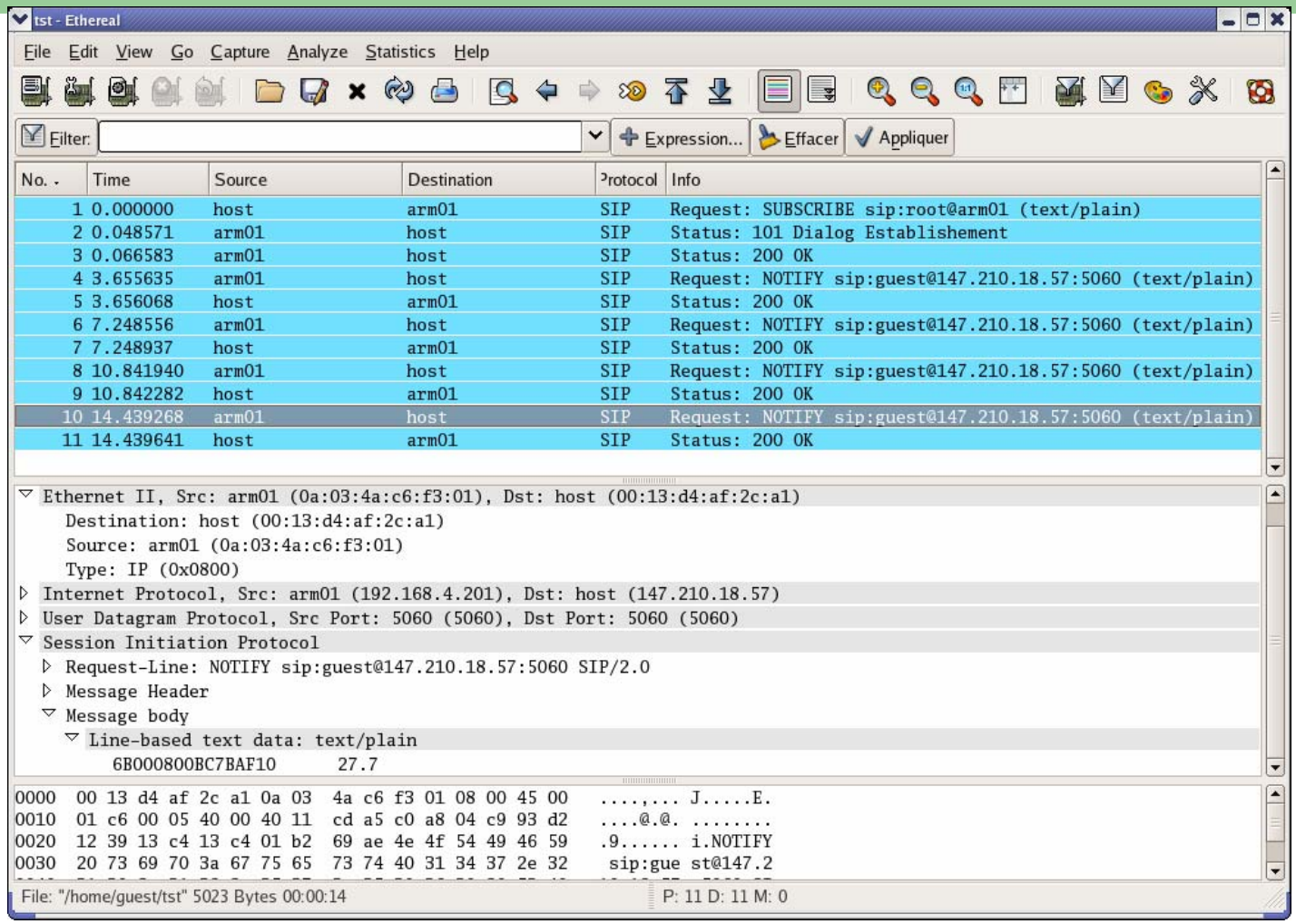
- Second case: SIP SUBSCRIBE and NOTIFY messages for ALARM events.

HomeSIP: SW development



SIP messages generated by send_notify

HomeSIP: SW development

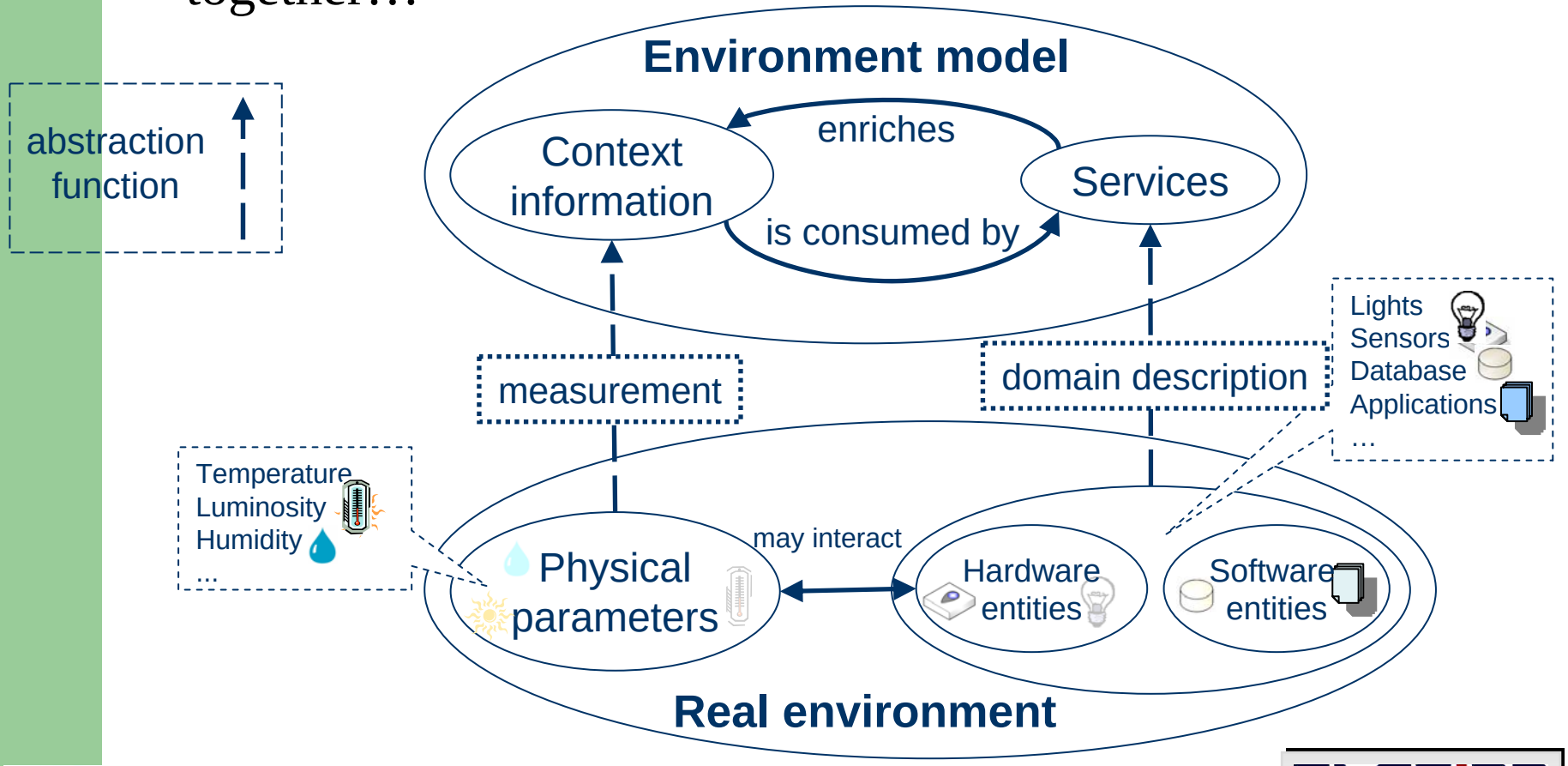


Traces generated by send_notify (27,7 °C)

THE NEED OF A DOMAIN SPECIFIC LANGUAGE

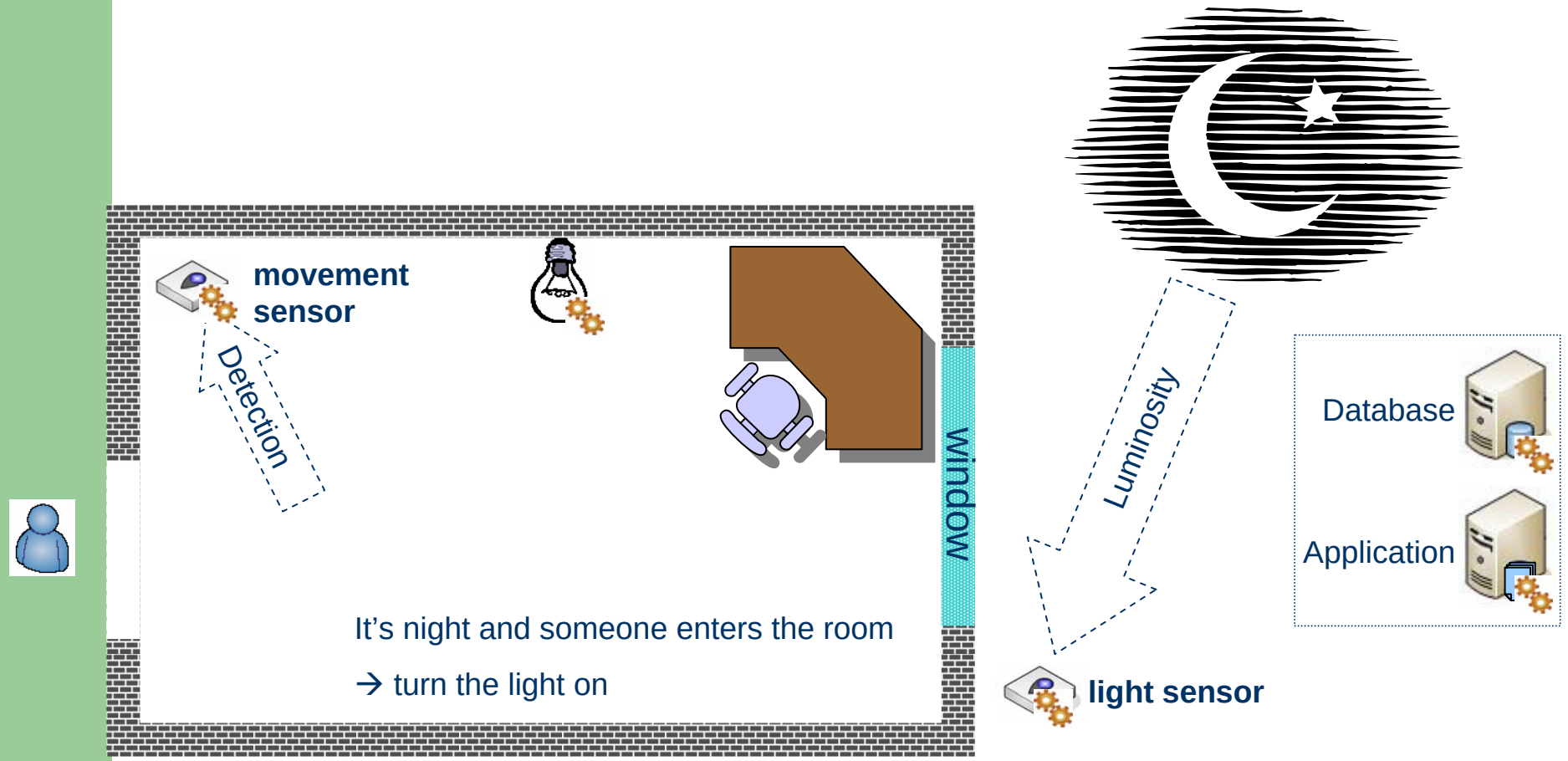
What is a pervasive Computing Environment?

- Pervasive computing environment is also known as ubiquitous computing: smart embedded devices are communicating all together…



An example of a pervasive Computing Environment

- A wide variety of entities with the associated context:



Why is challenging developing Applications?

- Pervasive heterogeneous entities:
  - Physical properties.
  - Exchanged data.
  - Interaction modes.
- Highly dynamic:
  - Appearing and disappearing services and users.
  - Deployment of new services.

Requirements

- Handling heterogeneity:
  - Describing and organizing services and exchanged data.
  - Defining the different interaction modes.
- Handling high dynamicity:
  - Managing runtime service reconfigurations.
  - Managing hardware dependencies.
- Functioning invisibly:
  - Robustness, reliability.
  - Minimizing interactions with users.

Comparison with existing Approaches

	Service Organization	Interaction modes	Data type	Approach
OSGI	Flat	**C**	Java type system	Middleware
UPnP	Flat	**C / E**	Schema XML	Middleware
Olympus	Ontology	**C**	Primitive type	Framework
WS umbrella	Ontology	**C / E / S**	User defined	Middleware
HomeSIP	Ontology	**C / E / S**	Hierarchical	DSL

C: Command, E: Event, S: Session

Solution: the DSL Approach

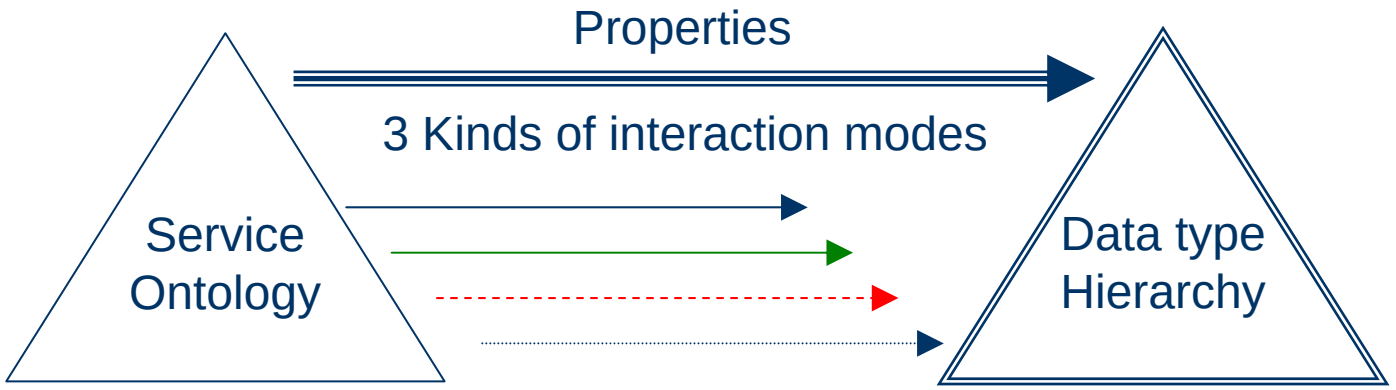
- DSL is the acronym of *Domain Specific Language*. A new language is defined (like C language) for a specific context:
  - Notion of service.
  - Ontology-based (tree approach).
  - Hierarchical data types.
  - Taxonomy of interaction modes.

- Benefits of the DSL approach:
  - Syntax and semantic enrichments.
  - Strong typing.
  - Runtime and compile-time verifications.

→ **Robustness and reliability**

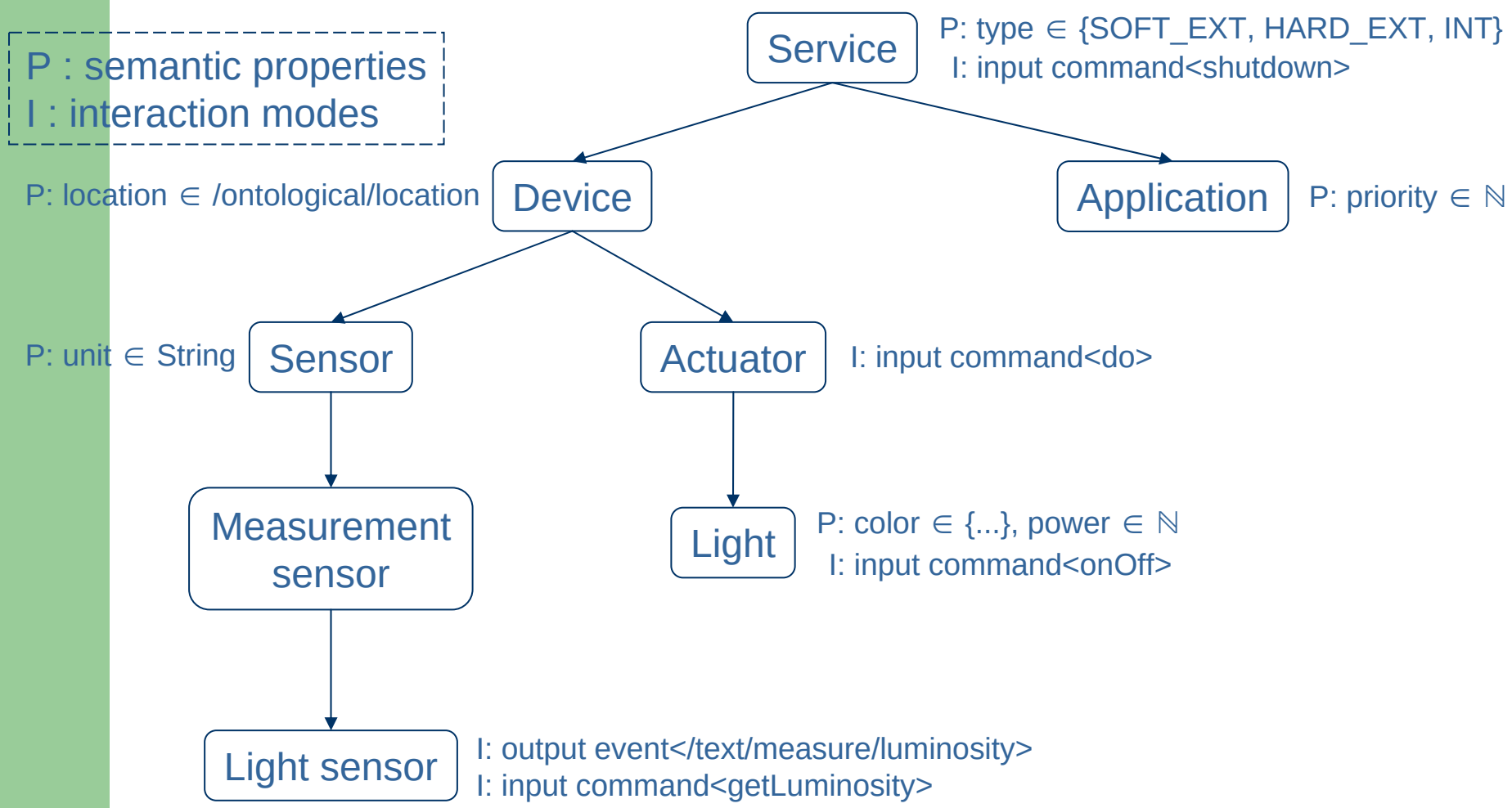
The Domain Description

- **data types: data structures for context information**
- **interaction modes: how to exchange complex data**
- **service ontology: service organization**

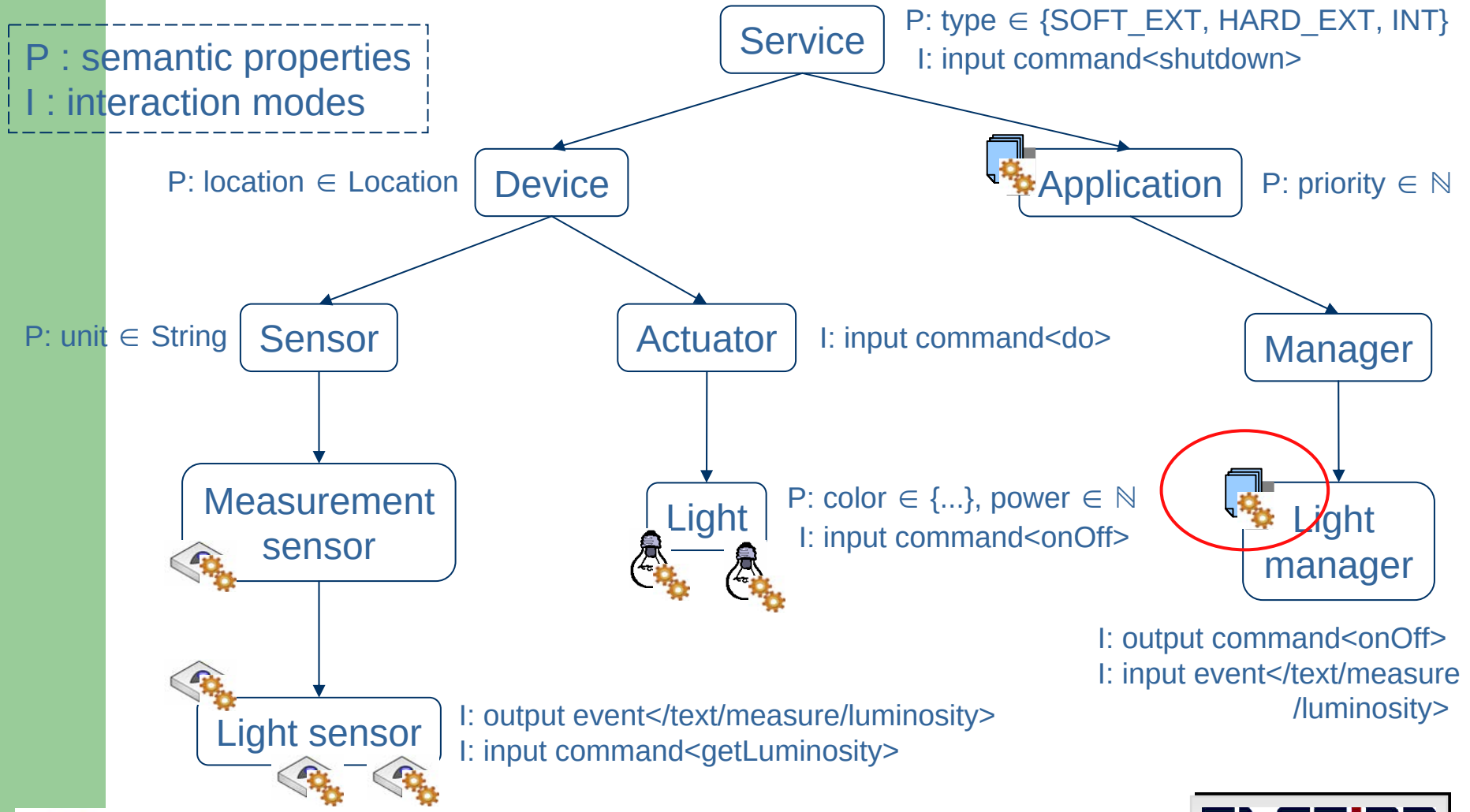


- Command (parameters and return value)
- Event (value embedded in the event)
- Session (values of the stream)

Example of Service Ontology



Example of Instances



CONCLUSION

Conclusion

- Launched in ’06, the HomeSIP project is operational.
- A minimal HomeSIP platform has been setup. Extensions with ZigBee network and PDA devices are under study.
- We have now SIP/sensor network gateways. We have embedded the GNU oSIP SIP stack.
- Different SIP scenarii have been written for PUT, GET and ALARM actions…We can consider this as firmware.

Conclusion

- We have designed and implemented a Domain Specific Language called SPL (Session Processing Language) to ease the development of telephony services first. SPL is currently extended for Home Automation.
- Some scenarii of typical services have been written in SPL and simulated.
- We have to join all software pieces together: DSL+firmware+hardware
- We have finally to test all in the real life!

Conclusion

Thank you for your attention!

Reminder

Special Embedded Session to the LSM ’07

Rencontres Mondiales du Logiciel Libre

10th to 14th july 2007 at Amiens, France

Call for Presentation

More info: http://www.rmll.info/

http://www.enseirb.fr/~kadionik/rmll2007/

References

- The HomeSIP project Page: http://www.enseirb.fr/cosynux/HomeSIP/
- The HomeSIP Phoenix Page: http://phoenix.labri.fr/research/pervasive/
- GNU oSIP SIP stack: http://www.gnu.org/software/osip/
- oSIP documentation: http://www.antisip.com/documentation/osip2/
- eXosip API: http://savannah.nongnu.org/projects/exosip/
- eXosip Documentation: http://www.antisip.com/documentation/eXosip2/

- Le projet HomeSIP : la domotique avec le protocole SIP. Hors Série numéro 25 Linux Magazine. Avril-mai 2006
- Source files available at: http://www.enseirb.fr/~kadionik/embedded/hs25/homesip.src.tar.gz
- The Cosynux Group Page: http://www.enseirb.fr/cosynux/
- The Phoenix Group Page: http://phoenix.labri.fr/