



## Prototyping home automation wireless sensor networks with ASSL

Emil Vassev, Mike Hinchey, Paddy Nixon

### Publication date

01-01-2010

### Published in

ICAC '10 Proceeding of the 7th International Conference on Autonomic Computing;pp. 71-72

### Licence

This work is made available under the [CC BY-NC-SA 1.0](#) licence and should only be used in accordance with that licence. For more information on the specific terms, consult the repository record for this item.

### Document Version

1

### Citation for this work (HarvardUL)

Vassev, E., Hinchey, M. and Nixon, P. (2010) 'Prototyping home automation wireless sensor networks with ASSL', available: <https://hdl.handle.net/10344/745> [accessed 25 Jul 2022].

This work was downloaded from the University of Limerick research repository.

For more information on this work, the University of Limerick research repository or to report an issue, you can contact the repository administrators at [ir@ul.ie](mailto:ir@ul.ie). If you feel that this work breaches copyright, please provide details and we will remove access to the work immediately while we investigate your claim.

# Prototyping Home Automation Wireless Sensor Networks with ASSL

Emil Vassev

Lero - The Irish Software  
Engineering Research Centre  
University College Dublin, Ireland

+353-1-716-5703

emil.vassev@lero.ie

Mike Hinchey

Lero - The Irish Software  
Engineering Research Centre  
University of Limerick, Ireland

+353-6-123-3607

mike.hinchey@lero.ie

Paddy Nixon

Lero - The Irish Software  
Engineering Research Centre  
University College Dublin, Ireland

+353-1-716-5361

paddy.nixon@lero.ie

## ABSTRACT

We target effective home automation based on wireless sensor networks. ASSL (Autonomic System Specification Language) is used to formally specify and generate prototype models for wireless sensor networks controlling a simulated virtual home environment. This approach allows for formal validation, experiments under simulated conditions, and smooth transition from a prototype system to a real one.

## Categories and Subject Descriptors

I.2.2 [Automatic Programming]: Program synthesis; Program transformation; I.5.5 [Implementation]: Special architectures

## General Terms

Design, Experimentation, Languages, Verification, Performance

## Keywords

wireless sensor networks, home automation, prototyping, ASSL

## 1. INTRODUCTION

Home automation (also called *domotics*) [1] is about integrating a computerized control system at home. Being an intelligent network of electronic devices, such a system is usually designed to monitor and control the home-integrated mechanical and lighting systems. From a technical perspective, a domotics system needs a computing infrastructure providing a composition of *controllers*, *sensors*, and *actuators*. It may employ sensor networks (SNs) (often wireless SN, or WSN) providing an infrastructure for monitoring, computing, and communication through a set of autonomous nodes equipped with sensors [2]. Ideally, a SN equipped with actuators may provide an efficient smart environment for home automation. With this project, we aim at research and development of *home automation wireless sensor networks* (HAWSNs). In our approach, a HAWSN is considered to be an *autonomic system* (AS) employing self-management by virtue of special policies driving the network in question. We rely on ASSL (Autonomic System Specification Language) [6, 7] to model and validate formally, and generate the implementation of prototype models for HAWSNs. Note that the possibility to formally validate HAWSNs is a considerable

advantage, because HAWSNs may be considered as critical systems where design or implementation flaws may be extremely expensive and even turn into hazards. With ASSL we develop prototypes where the needed sensors are generated as software stubs used to simulate runtime sensing of the environment. With this approach we can generate a series of prototypes to explore different aspects of a HAWSN and any successful prototype model may be transformed into a real HAWSN by employing hardware attached to the control software generated by ASSL.

## 2. BUILDING HAWSN WITH ASSL

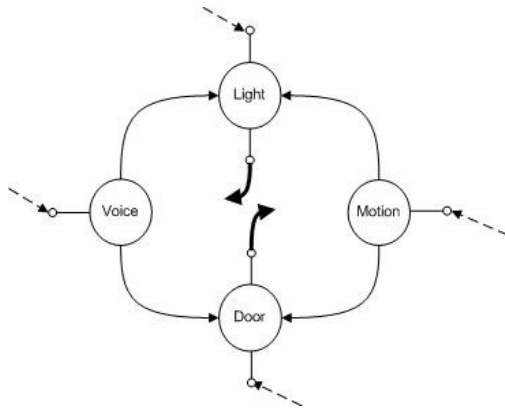
ASSL [3] is based on a specification model exposed over hierarchically organized formalization tiers (cf. Table 1). Each tier is intended to describe different aspects of the AS in question, such as *service-level objectives*, *policies*, *interaction protocols*, *events*, *actions*, *autonomic elements*, etc. This helps to specify an AS as being composed of special autonomic elements (AEs) interacting over interaction protocols. Usually an ASSL specification is built around self-management policies, which are driven by events and actions determined deterministically, similar to finite state machines. For the purpose of the HAWSN development, self-management policies may be specified to control sensors and actuators and exchange messages between the network nodes. Such policies can be specified at both the network-level and node-level. Thus, to specify a simple HAWSN we need to specify: 1) a single AE per sensor node providing the self-managing control software controlling the node's sensors and actuators, and 2) the communication with other AEs. Moreover, self-managing policies must be specified to provide self-managing behavior at the level of AS and at the level of AE. Note that although ASSL allows for specification and code generation of interaction protocols, to be appropriate for WSNs such a system needs to be empowered with the IEEE 802.11 protocol for wireless networks or Bluetooth. Here, we rely on ASSL to specify and generate intelligent sensor nodes in the form of AEs interacting over an embedded messaging system and on the IEEE 802.11 protocol (or Bluetooth) to help the messaging system connects these nodes wirelessly.

In the course of this project, we applied ASSL to develop a simple HAWSN model for a living room. With this model, we targeted lightening and door automation via voice commands, light sensors, and motion detection. Recall that with ASSL we specify intelligent network sensors as AEs. Here, we specified four distinct AEs composing the targeted HAWSN. They embed

managed elements helping to sense the environment, act, and communicate wirelessly:

- *Light AE* - controls the lights in the living room. It uses light sensors to determine the level of brightness in the room and a light switcher actuator to turn on/off the lights.
- *Voice AE* - controls microphones installed in the room. It processes input from the microphones to detect and recognize voice commands. It communicates with both Light and Door AEs that act on voice commands, such as “turn lights on/off” or “open/close door”.
- *Motion AE* - controls motion detectors to sense motion in the living room. It zones the living room and detects where the motion is taking place. This AE communicates with the Door AE and with the Light AE.
- *Door AE* - controls a door actuator to open or close the living room door. It also senses the area near the door for obstacles and communicates with Voice and Motion AEs.

Figure 1 depicts the data-control flow process in the prototyped HAWSN. Here, each of the AEs inputs data in the system via specific sensors (cf. dashed arrows in Figure 1). Next, AEs process that data locally and react by sending messages to other AEs (cf. thin arrows in Figure 1) or by acting in the home environment via its actuators (cf. bolted arrows in Figure 1). The Voice AE sends messages in the form of voice commands to both Light and Door AEs and the Motion AE sends messages in the form of relative coordinates of moving objects. Thus, both messages exchanged among AEs and data coming through sensors make AEs acting.



**Figure 1. Data flow in the prototyped HAWSN**

We specified the four AEs and generated their implementation with ASSL. These were generated with a special control loop that applies control rules specified and implemented as *self-management policies*. To evaluate the behavior of the prototyped HAWSN, we simulated sensing of the home-automation environment via ASSL events and predefined voice commands. With such events and voice commands we were able to simulate and experiment with *speech detection* and *voice-command recognition*, and make the Door AE react on voice commands. In order to demonstrate that the ASSL-developed prototype is an accurate implementation of the simulated HAWSN, we compared the runtime behavior of our implementation against the desired one. Data was gathered from *log records* produced automatically by the state-transition operations occurring in the prototype.

Table 1 shows some of the multiple experiments we performed with the HAWSN. The results are extracted from the log records.

**Table 1. Experiments with the HAWSN Prototype**

Test case	Simulated Conditions	Results
speech detection (Voice AE)	voice command “open door”	called interface function MICROPHONES.speechDetected; occurred event speechDetected; activated fluent inSpeech;
correct voice command (Voice AE)	voice command “open door”	test case “speech detection”; performed action recognizeCommand; called interface function MICROPHONES.retrieveCommand; called interface function WIRELESS_NETWORK.sendMessage; occurred event commandRecognized; terminated fluent inSpeech;
incorrect voice command (Voice AE)	voice command “open window”	test case “speech detection”; performed action recognizeCommand; called interface function MICROPHONES.retrieveCommand; occurred event commandNotRecognized; terminated fluent inSpeech;
open door (Voice AE & Door AE)	voice command “open door”; door closed	test case “correct voice command”; occurred as-level event mustOpenDoor; activated fluent inOpenDoor; performed action openDoor; called interface function DOOR.isDoorOpen; called interface function DOOR.open; occurred event doorOpened; terminated fluent inOpenDoor;

### 3. CONCLUSION

We have applied ASSL, an autonomic computing solution, to develop HAWSN prototypes where the system behavior is modeled as special policies controlling both the system’s sensors and actuators. We believe that the proposed approach helps developers decrease both development time and cost and increase quality of the targeted HAWSNs, because design and implementation flaws can be discovered and changes can be made while they are still inexpensive, i.e., before deploying a prototype.

**Acknowledgements.** This work was supported in part by an IRCSET postdoctoral fellowship grant (now termed as EMPOWER) at University College Dublin, Ireland and by the Science Foundation Ireland grant 03/CE2/I303 1 to Lero, the Irish Software Engineering Research Centre.

### 4. REFERENCES

- [1] Cheek, P. 2005. Aging Well with Smart Technology. In *Nursing Administration Quarterly*, vol. 29 (4), 329–338.
- [2] Lewis, F.L. 2004. Wireless Sensor Networks. In D.J. Cook and S.K. Das (eds), *Smart Environments: Technologies, Protocols, and Applications*. John Wiley.
- [3] Vassev, E. 2008. Towards a Framework for Specification and Code Generation of Autonomic Systems. Doctoral Thesis. Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada.