

# Services inside the Smart Home

## A Simulation and Visualization tool <sup>\*</sup>

Elena Lazovik,<sup>\*\*</sup> Piet den Dulk,<sup>\*\*</sup> Martijn de Groot,<sup>\*\*</sup> Alexander Lazovik,<sup>\*\*\*</sup>  
and Marco Aiello<sup>\*\*\*</sup>

{A.C.T.den.Dulk,M.de.Groote}@student.rug.nl,  
{e.lazovik,a.lazovik,m.aiello}@rug.nl

University of Groningen, Nijenborgh 9, 9747AG Groningen, The Netherlands

**Abstract.** Pervasive systems, and domotics in particular, is an application area where heterogeneity is the norm, with thousands of autonomous heterogeneous devices live together and need to interoperate. One of the greatest difficulties in developing middleware for smart homes is that this kind of systems are extremely difficult to test and verify. We propose to reduce the testing costs by replacing actual home services with virtual stubs behaving as if they were actual hardware installed somewhere in the house and, most importantly, to visualize the behaviour of the home to give the user the impression and feedback of being in a real home.

## 1 Introduction

Service-Oriented Computing (SOC) is a leading paradigm to create state of the art information systems. It is widely used in the development of complex software artifacts that require high interoperability, scalability, security, and reliability. Software of this kind can be seamlessly integrated with other systems that are possibly written in different programming languages and are deployed under different operating system. High interoperability is achieved through usage of standard platform-independent protocols, such as the web services stack. Pervasive systems and domotics applications are concerned with technology that pervades the home in order to make it more pro-active and aware with the final goal of increasing security and comfort of its inhabitants. SOC is a natural candidate to address the major requirements of domotics needs. Examples of are the Java based Jini technology (<http://www.jini.org>) or web services [2].

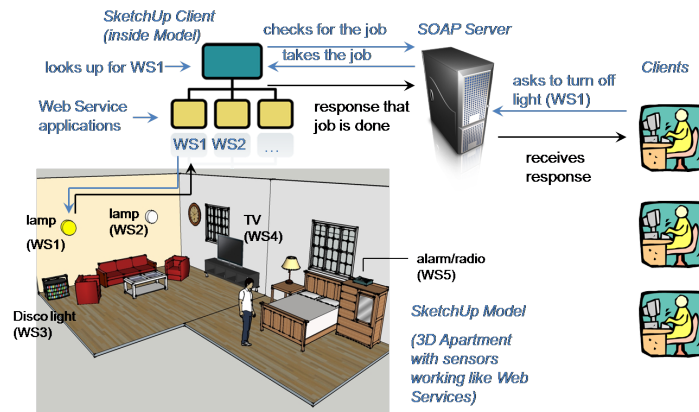
In this paper, we concentrate on smart homes, that is, homes that contain interactive and pro-active devices, that adapt their behavior to the needs of the home inhabitant through extensive interoperation and user interaction. For example, a movie may be automatically paused when the user leaves the room, and then launched again when s/he is back; windows are automatically opened

---

<sup>\*</sup> This research is supported by the EU through the STREP project FP7-224332 Smart Homes for All, <http://www.sm4all-project.eu>.

<sup>\*\*</sup> Master in Computer Science.

<sup>\*\*\*</sup> Distributed Systems Group



**Fig. 1.** Architecture of the domotics simulation environment ViSi.

to regulate the air condition or as a reaction to gas leak, and so on. One of the greatest challenges in domotics is verification of the proposed solutions.

We propose to reduce the testing costs by replacing actual home services with virtual stubs behaving as if they were actual hardware installed somewhere in the house. To this end, we extend Google SketchUp (<http://sketchup.google.com>) with a set of tools that extend its visual representation of the house with virtual interactive home web services supporting SOAP messages. This allows not only to visualize the potential smart home but also to provide a full featured simulation of any possible domotics scenario. In addition, it is possible to model the user and its interaction with the home. The realized simulation and visualization environment is named ViSi (Smart Home Visualization and Simulation) [1].

The framework may be used in conjunction with hardware supporting the web service stack. For instance, we have included in our test a controller of a fridge implementing WSDL and SOAP over HTTP on an ethernet connection. Figure 1 provides an overview of the implementation: the visualization component is written as a set of plug-ins for Google SketchUp, the communication mechanism is based on the Ruby SOAP implementation. As typical of web services, the clients are language-independent and can be written in any language that supports the web services stack. In our experimentation we have used Java and various BPEL engines. The implemented framework ViSi is not limited to domotics applications. Google SketchUp is a domain-independent drawing tool, and our framework may be used to simulate and visualize network applications, distributed systems, telecommunications, to name some application areas.

## References

1. ViSi demo. [www.sm4all-project.eu/index.php/activities/videos.html](http://www.sm4all-project.eu/index.php/activities/videos.html), 2009.
2. M. Aiello and S. Dustdar. A domotic infrastructure based on the web service stack. *Pervasive and Mobile Computing*, 4(4):506–525, 2008.