Using Formal Methods to increase confidence in one Home Network System implementation: Case study

Lydie du Bousquet[*], Masahide Nakamura[**], Ben Yan[***], Hiroshi Igaki[**]

[*]Universités de Grenoble, Laboratoire LIG, BP 72, 38402 Saint Martin d'Hères cedex, France
lydie.du-bousquet@imag.fr
[**]Graduate School of Engineering Science, Kobe University, Japan
{masa-n,igaki}@cs.kobe-u.ac.jp
[***]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
hon-e@is.naist.jp

Abstract. A home network system consists of multiple networked appliances, intended to provide more convenient and comfortable living for home users. Before being deployed, one has to guarantee the correctness, the safety and the security of the system. Here, we present the approach chosen to validate the Java implementation of one home network system. We relies on the Java Modeling Language (JML), to formaly specify and validate a model of the system. it.

1 Introduction

Emerging technologies enable general household appliances to be connected to LAN at home. Such smart home appliances are generally called networked appliances. A Home Network System (HNS) consists of multiple networked appliances, intended to provide more convenient and comfortable living for home users. Research and development of the HNS are currently a hot topic in the area of ubiquitous/pervasive computing. Several HNS products are already on the market (e.g. Hitachi (2007); Matsushita (2007); Toshiba (2007)).

A HNS provides many applications and services. They typically take advantage of wide-range control and monitoring of appliances inside and outside the home. Integrating different appliances via a network yields more value-added and powerful services (see Kolberg et al. (2003)), which we call HNS *integrated services*. For instance, orchestrating a TV, a DVD player, 5.1ch speakers, lights, curtains and an air-conditioner implements an integrated service, called *DVD theater service*, where a user can watch movies in a theater-like atmosphere.

For practical use of such services, it is essential to guarantee the correctness, the safety and the security of the services. A service should behave as specified (functional correctness). It must be free from the conditions that can cause injury or death to users, damage to or loss of equipment or environment (safety). And it must be protected against malicious adversaries to intrude or hijack the service (security). For instance, a *RemoteLock* service (that checks and locks doors and windows even from outside the home) must be disabled in case of a fire; otherwise a user might be locked into the room.