A Flexible, Plugin-Based Smarthome Simulation Framework with Application to Navigational Intention Prediction

John Staton and Manfred Huber
The University of Texas at Arlington

Abstract

The need for intelligent environments ("Smarthomes") to monitor and assist elderly individuals is on the rise and is projected to continue into the future. To assist in the development of autonomous Smarthome algorithms, a robust, flexible and extendable simulation environment is required. In this paper we present our approach to fulfilling this need through the development of a plugin-based simulation architecture, and include three different artificial intelligence based approaches towards the problem of predicting and modeling user intentions within a Smarthome environment.

Background

The term "Smarthome" is colloquially given to a place of domicile that includes some level of autonomous sensing and reaction. The environment of the home includes embedded sensors allowing for the "controller" to take in data (such as temperature, humidity, user positions, etc.) and perform some action or series of actions (turning on and off lights, changing the air conditioning or heating levels, monitoring, analyzing and reporting user actions and status) in response.

Smarthomes have an important role to play in the future of society. According to a 2005 study [Pollack, M. E. 2005], the percentage of the world's population over the age of 60 will reach 20% by the year 2050 due to increases in longevity and life expectancy from advances in healthcare and medicine. While this increase in life expectancy is a good development, it brings with it an increasing number of individuals living with one or more chronic condition and thus requiring special attention. Similarly, the prevalence of cognitive diseases, such as Alzheimer's has also been predicted by international organizations to increase significantly [Diamond, J. 2006] [Annual report. 2008-2009] [World Alzheimer Report. 2009].

Together this poses significant challenges for the health care and care systems, both in terms of needed care and of associated costs, which have to be handled.

To address these issues, the use of technologies, and in particular Smarthome technologies has been advanced in recent years [http://awarehome.imtc.gatech.edu][Helal, S.; et. al. 2008] and initial testbeds have been built and studied. In these studies, Smarthome technologies have generally been found to receive a positive reaction from older adults and have proven to be beneficial in preliminary trials [Demiris, G.; et. al. 2004] [Demiris, G.; et. al. 2008].

Another recent development in the area has been that the pervasiveness of Smarthome (or "home automation") hardware in the consumer market has increased significantly, with products ranging from thermostats, lighting systems, cameras, door locks, sensors, audio/visual systems and mobile app-based interfaces. Together, these trends have led to an increased market acceptance of the concept of home automation and a relative ease of finding sensors and connected hardware. On the other hand, the strong increase in available technologies has largely gone without a corresponding increase in "true", system level autonomous behavior[1], leaving this a fertile ground for academic Smarthome research to make these technologies more usable and to address the large amounts of data provided by the sensors.

To support and foster this significant potential for academic research and for applications in the Smarthome domain, an open-source, flexible simulation environment is needed to allow development, testing, comparison, and integration of different approaches and techniques. The only recent example of this is SIMACT from the Universite du Quebec at Chicoutimi [Bouchard, K.; et. al. 2012]. Their research showed the need for this type of project and that there was not yet an acceptable solution for it that could widely be used.

To address these needs of Smarthome research, this paper presents a framework for Smarthome simulations using a flexible Smarthome controller that facilitates the introduction of evaluation, prediction, and modeling components as plugins. Our approach emphasizes flexibility and extendability in order to simplify the integration and evaluation of new research components within the existing simulation.

[1] Individual component technologies with autonomous capabilities, such as the NEST Smart thermostat [http://www.nest.com], limited smart appliances, including LG's smart ThinQ devices [http://www.lg.com/us/discover/smartthinq/thinq], or monitoring technologies such as Sleep Number Bed's SleepIQ system [http://www.sleepnumber.com/sleepiq] for sleep monitoring, have become available but have so far stayed generally non-interoperable and thus cannot help establish system-level integrated behavior within the Smarthome environment.

Simulation Architecture

To provide a flexible development and test environment, we have developed a Smarthome simulation with a *plugin-style* architecture. The simulation takes care of the base functionality of managing the environment, objects, and users and communicates the state of these to the plugins.

It does this by utilizing instanced data managers, which simplify interaction with the plugins and guarantee that the data returned from the delegate functions is always correct (for instance, two different plugins will never receive two different user positions from the UserManager).

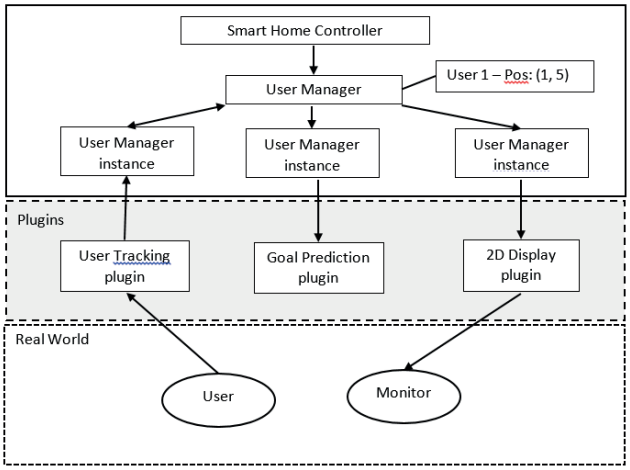

Figure1. Illustration of an instanced data manager interacting with multiple plugins

This design has numerous benefits. Firstly it removes the data structure and management requirements from the research and development efforts of scientists and allows them to focus on their specific novel domain. Secondly, it allows for direct comparison of two or more approaches; if the data management and underlying architecture remain the same across experiments, it acts as a control. Thirdly, this architecture can directly be applied to real-world Smarthomes by developing a plugin for hardware communication. Because each plugin is essentially a "black box" whose only connection to each other is through the managing architecture, as long as the translation from hardware to the simulation's base data structures is sound, all other plugins will still continue to function properly. Fourth, updating functionality of a single plugin will not accidentally prevent other plugins from functioning as normal, and this architecture allows for a significant amount of extended functionality through plugin development. Fifth, it allows for collaboration across laboratories, universities, and even scientific fields, because the architecture acts as an abstraction (a researcher does not have to know how to track a user in an environment to be able to use a User Tracking plugin).

In keeping with the open-source philosophy, the simulation has been created using Nokia's open-source QT windowing system in the C++ language (as opposed to Java, the language used in the SIMACT project) in the Windows operation system. QT is readily available, widely used in many consumer, industry and academic projects, and also portable to mobile devices. C++ is the third-most widely used programming language [Cass, S. 2014]. Also, as part of the base simulation, the 2D Graphics Display and GUI plugins are included, though not required to run the base Smarthome controller.

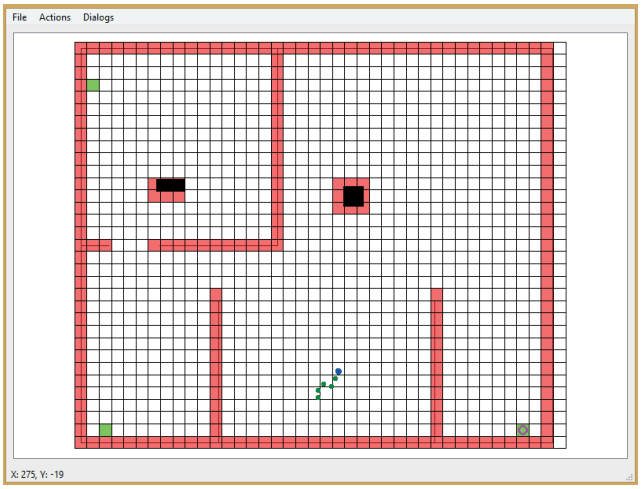

Figure2. Screenshot of the 2D Display plugin showing environment, goals, objects, user, recent user actions and predicted goal

Smart Features

To demonstrate the feasibility and flexibility of this approach, three autonomous functions have been implemented in the Smarthome simulation as plugins. In particular, three functions for user intention prediction in the context of navigation tasks are designed and implemented.

User Intention Prediction

First, we built on work done previously using harmonic function path planning as a method for modeling user behavior in a two dimensionally constrained environment [Staton, John H. C. 2008]. The desire is the ability to autonomously and in real-time predict the intended user destination within the Smarthome environment

This works by first taking a set of *a priori* goals in the environment (where a goal is an x-y location), discretizing the environment into a grid, and calculating the harmonic function over the grid, where obstacles (such as walls) equal 1, the goal equals 0, and all other grid locations obtain values between.

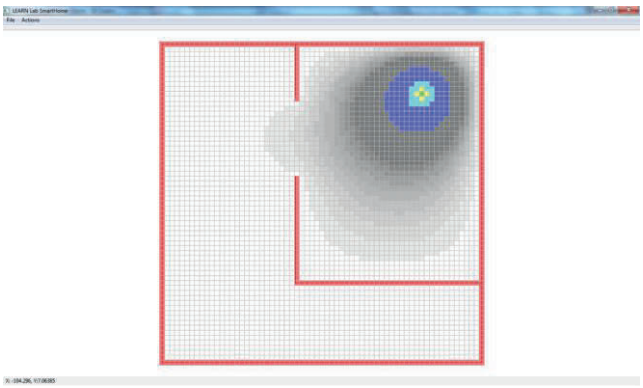

Figure 3. Screenshot of Smarthome simulation harmonic function result for single goal in simple environment

These functions form "paths" from the user's current location to each of the goals by "following" the downward slope of the function at each location in the grid. We then take a set of the user's recent "actions" (samples of the user's recent locations within the environment) and a forward *projection* of the user along their current trajectory and compare that to the generated harmonic function paths. Each action is also *weighted* to favor more recent actions over past actions. Finally, we include the prior probability of the goal (that is, the likelihood of that goal being the intended destination without considering any recent user action or projected future trajectory). All the parameters used are here represented as follows:

Set of recent user actions: $U_1, U_2, \ldots U_n$
Forward projected actions: $F_1, F_2, \ldots F_n$
Harmonic Function values: $H_1(U_1), H_1(U_2), H_1(F_1), \ldots$
Weighted values: $H_1(U_1) \times W_1, H_1(U_2) \times W_2, \ldots$
Prior Probability for Goal: $P_1, P_2, \ldots$

From these parameters, a goal ranking function can be determined which evaluates the degree to which a goal is consistent with observed actions and thus how likely it is to represent the navigation intention of the person:

$$G_1 = P_1 \times (H_1(U_1) \times W_1 + H_1(U_2) \times W_2 \ldots) + (H_1(F_1) \times W_1 + H_1(F_2) \times W_2 \ldots)$$

The result of applying this function to all available goal locations is a list of potential goals ranked by the probability of being the user's intended goal. This can then be used in the Smarthome simulation (or in a real world environment) to then perform an *action* or series of actions (such as opening a door or turning on lights).

As the predictions are performed over time, these goals prediction component also *learns* over time; as the user successfully reaches her or his chosen destination, the *a priori* probability of that goal, before user actions are taken into account, is modified; the more often the user goes to a certain goal, the more likely the plugin is to predict that goal and respond accordingly. This allows for the plugin to "grow" and modify itself along with the user and hone its behavior over time.

Sub-Goal Generation

With the result of the Goal Prediction, as described in the previous section, two goals may be predicted as equally likely. Instead of deciding between the two goals using some sort of stochastic method, we determined that there must be a point along the paths to both goals where they "branch". We refer to this "branch" point as a "sub-goal" because though it is not an end-goal like the previously defined *a priori* goals, it is significant enough from any non-goal point in the environment that it should be considered along the same lines as a regular goal.

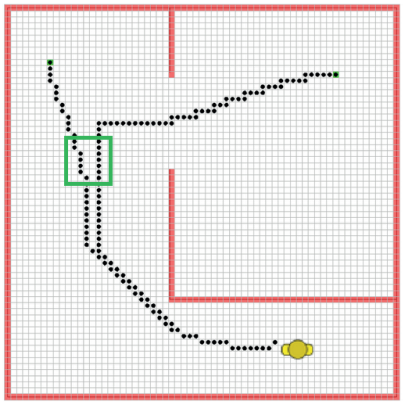

Figure 4. Visualization of sub-goal decision area

To determine where these sub-goals exist, situations where the top two ranked goals (from the Goal Prediction) are similar enough (within a modifiable threshold) are used. In these situations, the harmonic function paths for both potential goals (from current user location to end-goal location) are compared based on a similarity function. The x-y point where their similarity differentiates, i.e. where the paths towards both goals diverge from each other (again, within a modifiable threshold), is categorized as the subgoal. This sub-goal is then treated as if it were a normal goal until the user has approached and passed it, at which point the software has more information and can make a more informed prediction of the user's true intended goal.

Environmental Goal Generation

With predicting the user's intended goal destination in the environment and generating any necessary sub-goals implemented and functioning as intended, the question of where these goals should come from in the first place needs to be answered. Potential solutions include the user entering the goals manually or monitoring the user's behavior over a period of time and determining from their actions taken where

likely goals might be. However, both of those methods require some external effort to take place (the user entering the goals manually, or a training period of monitoring the user) and take time. This leads to a desire to develop and implement a system that could determine, at least at a basic level, initial goals on the first run.

To this end a plugin was developed that utilizes the environment through architectural pattern recognition.

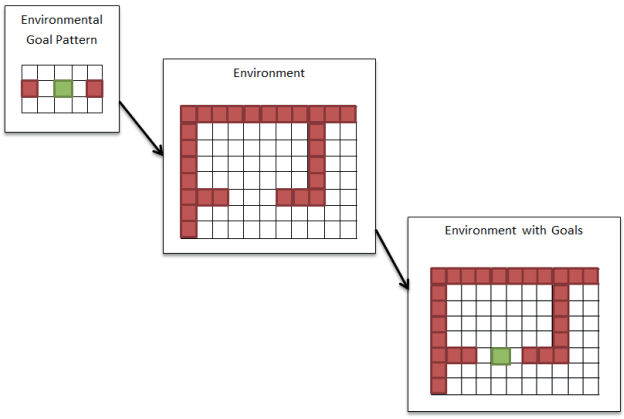

Figure 5. Example Goal Generation using a Door pattern

This plugin scans the environment using a collection of preset patterns and anywhere a pattern is matched in the environment, a goal is created. The basic idea here is to limit the need for pre-coding to the development of basic patterns and allowing the system to extrapolate this to occurrences of the pattern in the environment. For example, patterns such as general doorways, windows, or other points of interest can be relatively easily provided and then be automatically extrapolated to corresponding locations in the environment. Utilizing this, not only can potential goals be identified in a new environment that has not been seen before, this approach also allows detection of a wide range of potential targets with relatively low initial overhead. Once potential goals have been identified in this way, they can subsequently be utilized in the aforementioned Goal Prediction and Sub-Goal Generation plugins, and the Smarthome can then act in response to those predictions.

Conclusions and Future Work

This paper presents a flexible framework for the construction of Smarthome simulations built around a plugin framework for the easy integration and evaluation of various types of Smarthome data management, processing, and decision making components. Moreover, it describes a number of example plugins for the simulator which are aimed at navigational intention prediction. While this architecture has shown real promise in developing artificially intelligent Smarthome algorithms, more work needs to be done to bring it to a level where common usage could occur. This includes a three-dimensional simulation display, a scenario/script generator for proper experimentation, and an XML translator for communicating across a network with various hardware, at which point plugins to communicate with hardware can and should be developed.

References

Pollack, M. E. 2005. *Intelligent Technology for an Aging Population: The Use of AI to Assist Elders with Cognitive Impairement*. AI Magazine: Vol 26, No 2.

Diamond, J. 2006. *A report on Alzheimer disease and current research. Technical report, Alzheimer Society of Canada*. pp. 1-26.

Annual report. 2008-2009. *Alzheimer Society of Canada*. pp. 1-12.

World Alzheimer Report. 2009. *Alzheimer's disease International*. pp. 1-18.

Demiris, G.; Rantz, M. J.; Aud, M. A.; Marek, K. D.; Tyrer, H. W.; Skubic, M.; Hussam, A. A. 2004. *Older Adults' Attitudes Towards and Perceptions Of 'Smart Home' Technologies: A Pilot Study. Informatics for Health and Social Care*. Vol. 29, No.2, pp. 87-94.

Demiris, G.; Oliver, D. P.; Dickey, G., Skubic, M., Rantz, M. 2008. *Findings From a Participatory Evaluation of a Smart Home Application For Older Adults.* Technology and Health Care. Vol. 16, Number 2, pp. 111-118.

Bouchard, K.; Ajroud, A.; Bouchard, B.; and Bouzouane, A. 2012. *SIMACT: a 3D Open Source Smart Home Simulator for Activity Recognition with Open Database and Visual Editor*. International Journal of Hybrid Information Technology: Vol. 5, No. 3, July, 2012.

Cass, S. 2014. *Top 10 Programming Languages - Spectrum's 2014 Ranking*. IEEE Spectrum. Posted 19 Jul 2014.

Staton, John H. C. 2008. *An Assistive Navigation Paradigm For Semi-Autonomous Wheelchairs Using Force Feedback And Goal Prediction.* M.S. Thesis, Department of Computer Science, University of Texas at Arlington, Arlington, TX.

Helal, S.; Schmalz, M.; and Cook, D. 2008. *Smart home-based health platform for behavioral.*