A Multi-Agent System in Haskell

Tanja de Jong
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.dejong-4@student.utwente.nl

ABSTRACT

This proposal investigates a multi-agent home automation system in Haskell. As the name implies, a multi-agent system is a system that exists of multiple agents, that can individually solve problems through reasoning and communicate with each other in order to solve a bigger problem together. A home automation system integrates electrical devices in a house with each other in order to gain a higher quality of living for the residents. Because automation systems consist of multiple appliances and devices that are connected through a network and communicate with each other, a multi-agent system seems a good approach for the implementation of a home automation system. This system will be implemented in the programming language Haskell, which is a functional language. Functional languages are programming languages with a mathematical nature, which relates closely to the logical reasoning that is used by the agents of a multi-agent system.
This research will hopefully result in an answer to the question of how suitable functional languages are for the implementation of multi-agent systems.

Keywords

Multi-Agent System, Agent, Logic, Functional Language, Haskell, Home Automation System

1. INTRODUCTION

1.1 Multi-Agent System

A multi-agent system exists of multiple intelligent agents that can individually solve problems through reasoning and communicate with each other in order to solve a bigger problem together. Multi-agent systems can be used for solving problems that can not -or not as easily- be solved by a single agent or a monolithic system. These types of systems can have several advantages, for example:

- robustness;
- scalability;
- adaptability.

Because home automation systems consist of multiple appliances and devices that are connected through a network and communicate with each other, these types of systems are very complex to control [5]. For this reason, a multi-agent system seems to be a good approach to implementing a home automation system, which is the case that will be used for this research.

13^{th} Twente Student Conference on IT June 21^{st}, 2010, Enschede, The Netherlands.

1.2 Haskell

Functional languages are of a mathematical nature, and this might have advantages for implementing multi-agent systems [6]. Therefore, it seems to make sense to implement a home automation multi-agent system in a functional language. Because Haskell is an advanced and widely used functional language, the system to be implemented for this research will be implemented in Haskell.

1.3 Home Automation

Home automation is a popular subject nowadays, because of numerous advantages of automation that can add convenience and safety to people's lives. Therefore, a new generation of such systems with even more automation is called for.

Home automation is "a way of integrating technology and services in order to gain a higher quality of living", according to Stichting Smart Homes, Nationaal Kenniscentrum Domotica & Slim Wonen [7]. This is currently a very popular area of research as the number of new home installations are expected to increase enormously, according to Telecom research firm Berg Insight [3] and market research and intelligence firm ABI Research [1]. Nowadays, home automation systems can range from simple remote controls to complex networks with many different component that can each have their own form of intelligence and automation. The most intelligent systems that are currently in general use need user input through a pc, smartphone or tablet to start their automation processes. Automatation results in numerous advantages for people, for example:

- added convenience/fun/peace of mind: people have to perform less boring tasks themselves
- timesaving: automation does work that people would have to do otherwise
- saving money and better for environment: devices can be turned off automatically when nobody's home
- increased home safety: lights can be turned on automatically when somebody arrives home

1.4 Problem Statement

Multi-agent systems can be widely used for various applications, among which home automation, and can have many advantages in comparison to single-agent or monolithic systems.

Because logic can be used very well to reason about these multi-agent systems [**?**] and the mathematical nature of functional languages, it seems that a functional language would be a good choice for the implementation of a multi-agent system. Proposed here is to investigate how suitable a functional language is for the implementation of a multi-agent automation system, specifically, a home automation system in Haskell.

2. RESEARCH QUESTIONS

How suitable is Haskell for the implementation of a multi-agent automation system?

- What multi-agent automation systems exist in non-functional languages?
- How well does the mathematical nature of functional languages COMPLY with the logical reasoning of multi-agent systems

3. BACKGROUND

3.1 Multi-agent systems

"Multi-agent systems are systems composed of multiple interacting computing elements, known as *agents*.

3.1.1 Rational Agents

According to Wooldridge "an *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous* action in this environment in order to meet its design objectives." [9]. Thus, it is an entity that acts upon the environment it inhabits. An agent is said to be rational if it chooses to perform actions that are in its own best interests, given the beliefs it has about the world. Therefore, rational agents are expected to have the following properties, according to Wooldridge and Jennings [8]:

- autonomy;
- proactiveness;
- reactivity; and
- social ability.

3.1.2 The Belief-Desire-Intention Model

p.7 "The Belief-Desire-Intention (BDI) model gets its name from the fact that it recognizes the primacy of beliefs, desires, and intentions in rational action. Intuitively, an agent's beliefs correspond to information the agent has about the world. These beliefs may be incomplete or incorrect. An agents desires represent states of affairs that the agent would, in an ideal world, wish to be brought about. An agent's intentions represent desires that is thas committed to achieving". This model was originally developed by Michael Bratman [4] and contains three components:

- a philosophical component;
- a software architecture component; and
- a logical component.

3.1.3 Logic/reasoning

Why logic? - fixing on a structured, well-defined artificial language akes it possbile to investigate the question of what can be expressed in a rigorous, mathematical way - any ambiguity can be removed - by adopting a logic-based approach, one makes available all the results and techniques of what is arguably the oldest, richest, most fundamental, and best-established branch of mathematics Mathematical logic is one of the most important tools available in computer science and AI today. Logic has played a role in three aspects of software engineering, as:

- a *specification language*
  One of the main desirable features of a software specification language is that it should not dictate *how* a specification should be satisfied by an implementation. A logical specification language has exactly these properties. It does not dictate how an agent a should go about an intention, it is simply expected that a behaves as a rational agent given such an intention.
- a *programming language*
- a *verification language*

3.2 Haskell

4. RESEARCH METHOD

In order to answer the research questions, first of all, research will have to be done to find out whether comparable solutions in other languages exist and what the advantages and disadvantages of these languages are. In this case, comparable solutions denote implementations of various multi-agent automation systems that do not necessarily have the exact same functionality as the language that will be implemented. After this, the domain-specific embedded language can be implemented in Haskell. A method is needed on how to determine the performance of the implemented language and the other languages that it will be compared to and on how these performances can be compared. In order to determine the performance, the language will need to be simulated and tested. Then, the performance of the implemented language compared to already existing languages can be determined. In short, the research method consists of three parts:

- Performing research into existing comparable implementations in other languages
- Implementing a domain-specific embedded language in Haskell that can control a home automation system with a multi-agent approach
- Defining a method to determine and compare performances of these languages
- Determine the performance of the newly implemented language using simulation and testing
- Compare the newly implemented language to the existing implementations in other languages with the defined method

5. RELATED WORK

There is an extensive amount of research on home automation. However, most of this research does not focus directly on the topic of this proposal. The degree of automation most common in research needs user input in the form of using a smartphone, computer, or tablet to control the home automation system instead of having the intelligence to automate this. Besides this, there is a lot of focus on energy efficiency, for example by turning of devices when nobody is home [2]. Although this could ofcourse be accomplished by a higher degree of automation, the focus of the research mentioned in this proposal is on turning on devices when habitants return home, not on turning them off when they leave the house. [**?**]

6. RESEARCH SCHEDULE

Date	Description
13-04-2014	Research on other implementations completed
04-05-2014	Language implemented
11-05-2014	Method for comparison defined
18-05-2014	Performance of implemented language determined
25-05-2014	Comparison of languages completed
31-05-2014	Draft paper finished
13-06-2014	Paper finished

7. REFERENCES

[1] ABI Research. Home automation, security, and monitoring. 2012.

[2] A. S. al sumaiti, M. H. Ahmed, and M. M. A. Salama. Smart home activities: A literature review. 2014.

[3] Berg Insight. Smart homes and home automation. *M2M Research Series*, 2013.

[4] M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press: Cambridge, MA, 1987.

[5] G. Morganti, A. Perdon, G. Conte, and D. Scaradozzi. Multi-agent system theory for modelling a home automation system. 2009.

[6] A. Solimando and R. Traverso. Designing and implementing a framework forbdi-style communicating agents in haskell. 2012.

[7] N. K. D. . S. W. Stichting Smart Homes. The integration of technology and services in the home environment.

[8] M. Wooldridge. *Reasoning about Rational Agents*. The MIT Press: Cambridge, MA, 2000.

[9] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, Ltd, 2002.