Making Better Recommendations with Online Profiling Agents

Danny Oh, Chew Lim Tan
School of Computing
National University of Singapore
3 Science Drive 2
Singapore 117543
+65 6874 2900

dannyoh@digitaladept.com, tancl@comp.nus.edu.sg

ABSTRACT
In recent years, we have witnessed the success of autonomous agents applying machine learning techniques across a wide range of applications. However, agents applying the same machine learning techniques in online applications have not been so successful. Even agent-based hybrid recommender systems that combine information filtering techniques with collaborative filtering techniques have only been applied with considerable success to simple consumer goods such as movies, books, clothing and food. Complex, adaptive autonomous agent systems that can handle complex goods such as real estate, vacation plans, insurance, mutual funds, and mortgage have yet emerged. To a large extent, the reinforcement learning methods developed to aid agents in learning have been more successfully deployed in offline applications. The inherent limitations in these methods have rendered them somewhat ineffective in online applications. In this paper, we postulate that a small amount of prior knowledge and human-provided input can dramatically speed up online learning. We will demonstrate that our agent HumanE – with its prior knowledge or "experiences" about the real estate domain - can effectively assist users in identifying requirements, especially unstated ones, quickly and unobtrusively.

1. INTRODUCTION
Electronic profiling agents have been deployed with considerable success in certain agent-based recommender systems, namely, information filtering (IF) systems, collaborative filtering (CF) systems and hybrid recommender systems.

Unfortunately, the successes of these systems have been restricted to simple consumer goods such as movies, books, clothing and food. When the IR and/or CF techniques plus other reinforcement learning methods are applied in online applications for complex consumer products such as real estate, vacation plans, insurance, mutual funds, and mortgage, they fail to enjoy much success.

This is because many reinforcement learning implementations assume that the agent developed knows nothing about the environment to begin with, and that the agent must gain all of its information by exploration and subsequent exploitation of learned knowledge.

When dealing with a real, complex online system such as a large-scale real estate listing and brokerage application, however, this approach is simply not practical. Typically, the state space is too large to explore satisfactorily within the lifetime of the agent (much less within the attention time-span of typical online users).

Worse still, making "random" exploratory recommendations can frustrate and disappoint the user, potentially causing the user to abandon the system totally.

Accumulated knowledge in the form of memories and experiences allows humans to go about performing daily tasks. In the real world, we often go to a human real estate agent for assistance in selling or acquiring real estate properties. We naturally expect the agent to be an expert in the real estate domain, and hence able to offer suitable advice and recommendations. Certainly, we do not expect the real estate agent to have no knowledge about the real estate domain.

Hence, in order to take our prior knowledge (which are often implicit) and incorporate them into a reinforcement learning framework, we have examined in this work the idea of supplying the agent with an initial policy about the real estate domain in the HumanE agent ("HumanE").

2. BETTER RECOMMENDATIONS WITH HUMANE

2.1 Difficulties in Developing Profiling Agents for Complex Domains
The main problems encountered when developing online profiling agents for complex multi-dimensional domains can be summarized as below:

- Assumption that the agent knows nothing and must acquire its knowledge through exploration and subsequent exploitation of learned knowledge results in slow agent learning for complex domains and makes online implementation difficult

- Difficult to give an agent large amount of application-specific and domain-specific knowledge
- Difficult to encode this knowledge in an abstract language
- Difficult to transfer agent knowledge and the control architecture for building agents for other applications
- Difficult to maintain the individual rules in the agent rule base over time
- Static agent knowledge (i.e. cannot be customized to individual user habits and preferences)
- Making "random" exploratory recommendations can frustrate and disappoint the user
- Difficult to allow for serendipitous discoveries of user preferences
- Difficult to obtain user trust when an interface agent is very sophisticated, qualified and autonomous from the start
- Too much data is required in an online setting for typical learning methods (e.g. reinforcement-learning methods)

2.2 Practical Approach

We strongly believe that practical agent learning for online applications is possible by integration with human-supplied knowledge. This is because humans can provide a lot of help to assist agents in learning, even if humans cannot perform the task very well. Humans can provide some initial successful trajectories through the space. Trajectories are not used for supervised learning, but to guide the learning methods through useful parts of the search space leading to efficient exploration of the search space.

Online profiling agents can be bootstrapped from a human-supplied policy which basically gives some sample trajectories. The purpose of the policy is to generate "experiences" for the agents. This policy can be hand-coded by domain experts. It need not be optimal and may be very wrong. The policy shows the agents "interesting" parts of the search space.

Our online profiling agent, HumanE, is based on the aforementioned approach and it offers users the opportunity to find products that will best meet their requirements. HumanE guides users through a product selection process. Users get to specify information about their individual requirements and restrictions by creating and refining their profiles.

Based upon the profile (and initial policy if the profile is newly created), HumanE offers an initial selection of products. Users can then select from these matching products to view more detailed product information such as product features. HumanE also tries to be helpful by providing products that are newly added as well as products that are popular among other users.

To refine the profile, users can specify which features are desirable or undesirable through an intuitive and friendly interface, and HumanE will offer a new selection of products matching the revised profile. If no matching products are found, users can backtrack to their previous profile.

2.3 Design Assumption

Our design approach assumes that the entire user experience is an iterative process of browsing and meaningful user feedback. The approach has in fact been adopted successfully by similar systems such as RentMe [2, 3], CASA [7] and Apt Decision [12]. As the user is actively involved throughout the entire profile creation process, the user can react independently to every feature of the real estate offerings.

2.4 Domain Analysis

To test the feasibility of the proposed learning model, we chose the real estate domain. As the agent needed to have built-in knowledge about the domain, we analyzed online and offline apartment advertisements to determine the standard apartment features for the local real estate domain.

Next, we considered how people choose apartments. After examining the features, we concluded that some of them (e.g. district, type, price) were pivotal to the final choice of apartment. That is, most people would reject an apartment if the value for a crucial feature were not to their liking. Other features (e.g. bridge, underpass, swimming pool) were less pivotal. All this domain knowledge went into HumanE.

In addition, we examined two destinations of apartment seekers: real estate websites and human real estate agents, to determine what knowledge we could glean from those interactions.

2.4.1 Real Estate Websites

Many real estate websites adopt either the pure browsing metaphor [14] or the search-like metaphor [13]. One problem is that users are expected to enter many specific details about their ideal apartment. Another problem is that they must enter their preferences when they visit a new site and each time they visit the site. This is because there is no option to save multiple sets of preferences for a single site.

2.4.2 Humane Real Estate Agents

We consider how people deal with the ambiguity and imprecision of real world decisions. For example, when a customer interacts with a real estate agent, the customer may react in a variety of ways not limited by answers to explicitly posed questions. The agent's description will typically contain many details not asked for originally by the customer.

The success of the interaction is determined largely by the agent's ability to infer unstated requirements and preferences from the responses. Near-miss examples establish whether the ostensible constraints are firm or flexible. Good agents are marked by their ability to converge quickly on a complicated set of constraints and priorities.

2.4.3 Transferring Domain Knowledge

Much of the work done for HumanE would transfer well into any domain in which the user could browse the features of a complex object. That is, objects such as calling plans, mutual funds, homes, computers, vacation plans, or cars would work well, but simple consumer goods such as clothing or food would not.

Transferring the agent into another domain would require the services of a subject matter expert who could identify salient features of the complex objects in the domain, alter the program to work with those features and determine which features were crucial to the final decision. After testing on a suitable list of objects, the "new" agent could be released.

3. LEARNING APPROACH

3.1 Introduction

The proposed two-phase learning approach has been tested successfully in past research on robotics [16].

Kaelbling et. al. found that robots using reinforcement learning learnt better when they were provided prior knowledge about their environment using a supplied initial policy. The policy generated example trajectories through the state-action space and showed the robot areas of high rewards and low rewards. After the robot had acquired a suitable amount of information through this initial phase of learning, the reinforcement learning system took control of the robot. Usually by this time, the robot had learned enough to make more informed exploration of the environment.

In this work, we adapt a similar approach when building an agent-based online real estate system. To do so, we consider each user decision as a trajectory in the search space much like the trajectories in the robot motion.

3.2 Initial Profile vs Initial Policy

Initial profile refers to the profile that is created at the very beginning of the learning approach. The initial profile contains only the user-defined preferred district, desired apartment type, and price.

Initial policy refers to the set of trajectory samples that show HumanE areas of high rewards and low rewards in the search space.

3.3 Constituents of a Profile

The main objective of HumanE is to create a user profile (or simply called a profile) to store user preferences and to assist the user to refine his or her profile intelligently using the supplied learning approach.

In our scenario, a profile stores both static and dynamic (learned) user preferences in the form of desired and undesired apartment features. Examples of apartment features include "high floor", "near MRT (Mass Rapid Transit)", "marble floor", etc.

3.4 Overview of the Learning Approach

We have adopted a two-phase learning approach for HumanE.

In the first phase of learning, HumanE learns by reinforcement, observation and takes actions that arise from a supplied initial policy. This mode of learning will last for one iteration of the profile refinement process (i.e. the iterative process of viewing apartments and selecting/ranking desired and undesired features).

In the second phase, HumanE learns by reinforcement and observation. The content of the initial policy is dynamic as it is updated without human intervention from the actions taken by HumanE.

Figure 1 depicts the workflow of the two-phase learning approach.

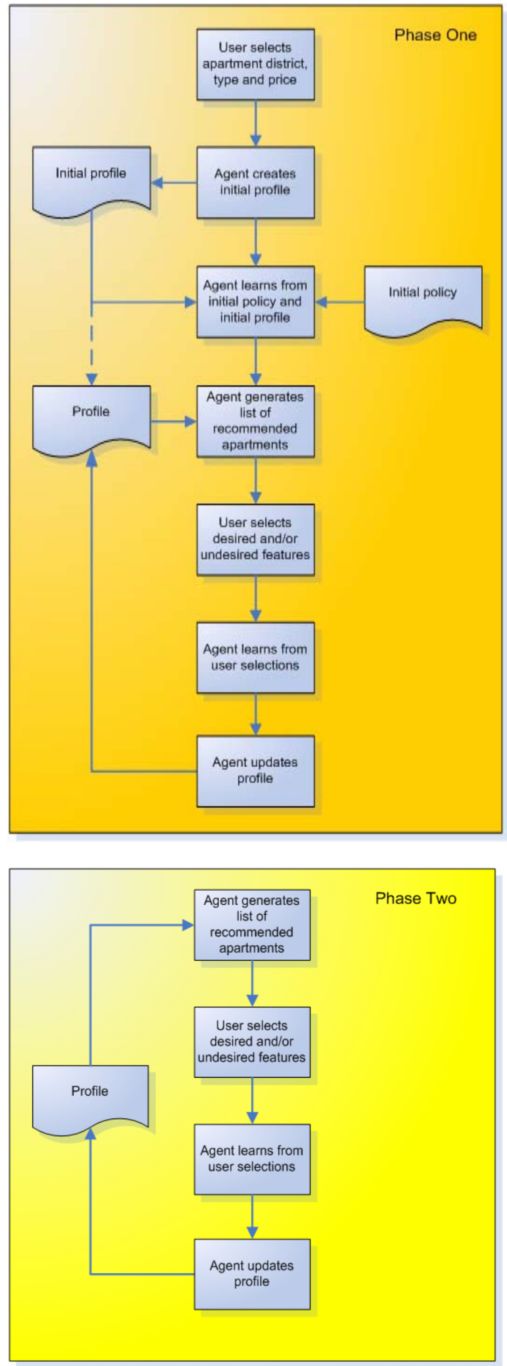

Figure 1. Workflow for the two-phase learning approach

3.5 Phase One Learning

In phase one learning, HumanE initially learns and takes action from a supplied initial policy. This occurs right after when an initial profile is created. Using the information stored within the supplied initial policy and initial profile, HumanE learns the locations of desired (high rewards) and undesired (low rewards) apartments in the search space that match the user preferences stored in the initial profile.

3.5.1 Learning from an Initial Policy

To emulate some of the inference power a human real estate agent might have, we have incorporated an initial policy to enhance the interactive learning ability of the agent. Basically, an initial policy is a XML file that stores information about which apartment features are generally considered as desirable and undesirable.

In HumanE, the initial policy is stored in a file named “Bootstrap.xml”. Figure 2 shows the content of this file.

```
<?xml version="1.0" encoding="utf-8" ?>
<features>
  <item id="desired" value="1,4,8,98"/>
  <item id="undesired" value="23,24,43"/>
</features>
<desired>
  <district>
    <item id="1"
value="15,45,92,123,280,410,488,523,677,712"/>
    ……
  </district>
</desired>
<undesired>
  <district>
    <item id="1"
value="23,34,52,166,232,359,390,416,509,682"/>
    ……
  </district>
</undesired>
```

Figure 2. Initial policy used in HumanE

The first piece of information encoded in the initial policy as shown above is the list of features generally considered as desirable and undesirable features. The attribute “id” denotes the attribute name and the attribute “value” refers to the value of the attribute “id”.

The second piece of information encoded is the list of top ten most popular apartments per district. The attribute “id” denotes the attribute name which in this case refers to the district id. The attribute “value” refers to the value of the attribute “id” i.e. the apartment id of the top ten most popular apartments per district.

The third piece of information encoded is the list of top ten most unpopular apartments per district. The attribute “id” denotes the attribute name which in this case refers to the district id. The attribute “value” refers to the value of the attribute “id” i.e. the apartment id of the top ten most unpopular apartments per district.

Coupling with the information stored in the initial profile, the initial policy will generate certain “interesting” trajectories through the search space, showing HumanE areas of high (popular apartments) and low (unpopular apartments) rewards. These trajectories and associated rewards are then used in this first, passive phase of learning.

Therefore, HumanE is able to generate a larger but potentially interesting set of matching apartments. This gives the user an opportunity to learn about apartments that do not exactly match the initial profile but may be of interest to him. In this way, HumanE allows for the serendipitous discoveries of new user preferences without the danger of random, unguided “exploratory” recommendations.

As shown in Figure 1, the bootstrapping process occurs after the creation of the initial profile. Here, HumanE observes the states and rewards being generated and bootstraps this information into its “memory”. In our domain, the areas of high rewards are those apartments which have at least half of the “desired” features as specified in the initial policy.

On the other hand, the areas of low rewards are those apartments which have at least half of the “undesired” features as specified in the initial policy. We have labeled each feature as either “desired” or “undesired” based on commonsense rules.

In short, this corresponds to a real-life situation in which a human real estate agent always has in mind a small number of real estate properties that he or she knows that are popular or unpopular by virtue of the features they had. The initial policy will show HumanE where the locations of potentially “desired” and “undesired” apartments based on the profile.

For example, if the user has specified in the profile that the features desired are “MRT”, “Schools” and “High Floor”, then HumanE will search for matching apartments that have a combination of the following criteria:

- All desired features as stated in the profile (“MRT”, “Schools” and “High Floor”).
- More than half of the desired features as stated in the initial policy (e.g. “MRT”, “Bus Stop”, “Lift Level”, “Mid Floor”, “Good View”, and “Windy”).
- Less than half of the undesired features as stated in the initial policy (e.g. “Playground”, “Rubbish Dump”, “Low Floor”, “Blocked View”, and “Facing Afternoon Sun”).

3.5.2 Reinforcement Learning using a Multidimensional Utility Function

After HumanE has generated a list of recommended apartments, it adopts reinforcement learning as the next learning technique to learn user preferences.

It learns a multidimensional utility function on states (or histories) and uses it to select actions that maximize the expected utility of their outcomes. The reinforcement learning approach is used through the entire profile refinement process.

In this way, the multidimensional utility function is able to capture past profile changes (i.e. the agent remembers history information) and incorporate the knowledge learned into a simple representation to be used by the matching algorithm.

3.5.3 Learning by Observation

To augment the serendipitous discoveries of apartments which can be of potential interest to the user, we have implemented the “favourites” and “views” functions. First, the user can specify an apartment to be added to a “favourites” list for a particular profile.

Second, the user can select an apartment to develop the profile or simply to view more details about it.

As part of the matching process, HumanE selects the top ten apartments which are the top "favourites" (Figure 3) and widely "viewed" (Figure 4). This encourages the user to make more serendipitous discoveries of apartments which the user may be interested in. The assumption taken here is that there is a high possibility that a typical user may be interested in apartments which are generally considered by other users to be "good".

MOST POPULAR APARTMENTS ::.::.:..

Street/Block/Unit	District	Type	Area	Price	#Favs	#Views		
Blk 656 Jln Tenaga #05 and above	Bedok / Chai Chee	5I	0	$320,000.00	98	80	[Fav]	[Details]
Blk 51 Bukit Merah View	Bukit Merah / Queenstown / Telok Blangah	4S	0	$420,000.00	98	49	[Fav]	[Details]
Blk 30 Jalan Klinik #03	Bukit Merah / Queenstown / Telok Blangah	3	0	$335,000.00	98	50	[Fav]	[Details]
Blk 219 Spottiswoode Park Rd #02	Choa Chu Kang / Bukit Panjang	3A	0	$320,000.00	98	94	[Fav]	[Details]
Blk 485A Choa Chu Kang Ave 5 #08	Choa Chu Kang / Bukit Panjang	4A	0	$320,000.00	98	71	[Fav]	[Details]
Blk 112 Hougang Ave 1 #04 and below	Hougang / Serangoon / Yio Chu Kang	4NG	0	$380,000.00	98	90	[Fav]	[Details]
Blk 510 Hougang	Hougang / Serangoon / Yio Chu Kang	5I	0	$245,000.00	98	29	[Fav]	[Details]
Blk 128 Lorong Ah Soo	Hougang / Serangoon / Yio Chu Kang	EA	0	$320,000.00	98	43	[Fav]	[Details]
Blk 101 Serangoon North Ave 1 #10 and above	Hougang / Serangoon / Yio Chu Kang	4	0	$290,000.00	98	29	[Fav]	[Details]
Blk 317 Tampines	Tampines / Simei	5I	0	$170,000.00	98	96	[Fav]	[Details]

Figure 3. Most Popular Apartments

MOST VIEWABLE APARTMENTS ::.::.:..

Street/Block/Unit	District	Type	Area	Price	#Favs	#Views		
Blk 130 Clarence Lane	Bukit Merah / Queenstown / Telok Blangah	5	0	$480,000.00	55	98	[Fav]	[Details]
Blk 52 #08 Kent Rd	Central / City	4NG	0	$210,000.00	38	98	[Fav]	[Details]
Blk 51 #10 Kent Rd	Central / City	4NG	0	$201,000.00	89	98	[Fav]	[Details]
Blk 287 Choa Chu Kang	Choa Chu Kang / Bukit Panjang	5EM	0	$400,000.00	64	98	[Fav]	[Details]
Blk 14 Clementi	Clementi / West Coast	5S	0	$250,000.00	19	98	[Fav]	[Details]
Blk 8 Jalan Batu #09	Marine Parade	3	0	$400,000.00	48	98	[Fav]	[Details]
Blk 185 Pasir Ris St #09 and below	Pasir Ris / Changi	5I	0	$255,000.00	51	98	[Fav]	[Details]
Blk 867A Tampines	Tampines / Simei	5EA	0	$375,000.00	19	98	[Fav]	[Details]
Blk 655 Woodlands Ring Rd	Woodlands / Sembawang	5I	0	$220,000.00	8	98	[Fav]	[Details]
Blk 128 Marsiling Rise #06 and above	Woodlands / Sembawang	EA	0	$260,000.00	33	97	[Fav]	[Details]

Figure 4. Widely Viewed Apartments

3.5.4 *Matching Algorithm*

HumanE employs a matching algorithm that is based on the concept of property relaxation. It uses a sequence of built-in rules when searching for matching apartments. If no matching apartments can be found, HumanE displays the apartments listed in the "top ten most popular apartments per district" information contained in the initial policy for the district specified in the profile.

3.5.5 *User Interface*

We do not expect an average user to have a high degree of computer skills. Hence, we have paid extra attention to the design of the agent interface. We have made several changes to the HumanE agent interface in the hope that the interface will be intuitive and responsive to users' actions.

After trying out a few approaches and gathering some useful user feedback, we decided to use two list boxes i.e. "desired" list box and "undesired" list box to allow the user to specify which feature is desirable or undesirable. Figure 5 shows the interface of the current version of HumanE.

The user clicks on the left arrow button to add a feature to the "desired" list box or clicks on the right arrow button to add a feature to the "undesired" list box. Furthermore, the user can rank the features in each list box in accordance to their liking. The features at the top of each list box correspond to those features well-liked most or dislike most.

One advantage of using the present user interface is that the user is "compelled" to consider carefully their liking of the features as indicated in the list boxes and this may help to speed up the process of discovering unstated user preferences.

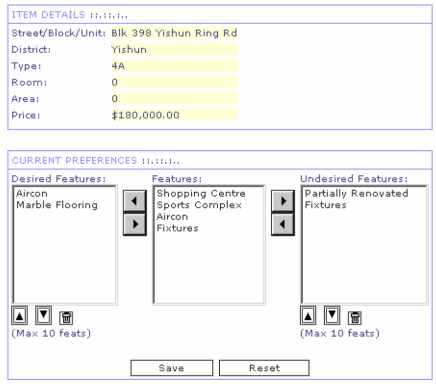

Figure 5. Current version of HumanE interface

3.6 Phase Two Learning

The purpose of having the initial policy in phase one learning is simply to generate experiences of the world which is tantamount to incorporating prior knowledge into HumanE. After a suitable amount of information has been acquired in the bootstrapping process, the second phase of learning takes over where HumanE learns primarily using reinforcement learning and learning by observation. Usually by this time, HumanE is more "knowledgeable" which allows for more informed exploration of the search space.

3.7 Benefits of Proposed Learning Approach

One of the main reasons why many reinforcement learning implementations fail to achieve much success for complex goods is that it is assumed that the agent developed knows nothing about the environment to begin with and that the agent must gain all of its information by exploration and subsequent exploitation of learned knowledge.

When dealing with a real, complex online system such as a large-scale real estate listing and brokerage application, however, this approach is simply not practical. Typically, the search space is too large to explore satisfactorily within the lifetime of the agent (much less within the attention time-span of typical online users). Worse still, making "random" exploratory recommendations can frustrate and disappoint the user, potentially causing the user to abandon the system totally.

For example, Apt Decision [12] suffers from the possibility that the user may get frustrated and disappointed if no suitable recommendations are found during the initial use of the system. This scenario can result as the Apt Decision agent has no prior knowledge about the real estate domain and cannot make good recommendations initially. Moreover, the agent needs time to learn the user's preferences from scratch and the time taken could be significantly long enough to cause the user to give up on the agent.

Another example is the SurfJet Agent [15] which is an intelligent assistant (much like HumanE) that acts as an autonomous learning agent. It is non web-based and uses an interest profile to perform searches on the Internet for articles on the user's behalf. SurfJet is able to make more accurate and useful searches as compared to traditional searching techniques as the user can give it a profile describing many of his or her interests, including how interesting (or uninteresting) each item is and how they relate to each other.

However, SurfJet does not store any prior knowledge and rely solely on the iterative process of user feedback and profile refinement to make increasing accurate recommendations. Making good recommendations in the early stages of learning could be difficult and, like Apt Decision, a considerable amount of time may be spent to train SurfJet to understand a user's stated and unstated interests. It is likely that many users may not be prepared to commit that kind of time and effort to train the agent until it is sufficiently capable of making fairly good recommendations.

Accumulated knowledge in the form of memories and experiences allows humans to go about performing daily tasks. In the real world, we often go to a human real estate agent for assistance in selling or acquiring real estate properties. We naturally expect the agent to be an expert in the real estate domain, and hence able to offer suitable advice and recommendations. Certainly, we do not expect the real estate agent to have no knowledge about the real estate domain.

Hence, in order to take our prior knowledge (which are often implicit) and incorporate them into a reinforcement learning framework, we have examined the idea of supplying HumanE with an initial policy about the real estate domain and in this section, we have briefly described the learning approach which we are confident that it can aid profiling agents in making better recommendations faster with the ultimate aim of soliciting greater satisfaction, confidence and trust from users. We will support our claims using experimental results and the details can be found in the next section.

4. EXPERIMENTAL ANALYSIS

4.1 Methodology

The following sections outline the methodology used for the experiments conducted with HumanE and our testers.

4.1.1 Metrics

There are four dimensions to measure HumanE's ability to increase customer satisfaction:

- Number of profile changes
- Time taken to create a profile
- Ease of use
- Performance

4.1.2 Test data

To ensure the realistic nature of the experiments to be conducted, we painstakingly created our test data set from more than eight hundred actual real estate ad postings from both offline and online sources.

4.2 Experimental Design

Basically, we want to test whether our proposed learning approach with the use of initial policy contributes to better performance for web profiling agents. Based on the findings in [8, 9], we decided to use survey research in our experimental design. In addition, our experimental design is also strongly influenced by the findings from [6, 10] especially in the area of web page evaluation.

We invited one hundred and fifty genuine real estate buyers to evaluate HumanE based on the chosen metrics. Most testers were of the age between late twenties to fifty years old. This coincided with the age range where most people would consider buying apartments and would genuinely be interested in using HumanE as an intelligent assistant during the searching and selection of apartments.

We also took into consideration the occupational profiles of the testers. We wanted to avoid having many IT professionals as our testers and they would naturally be more IT-savvy and might be inclined to rate HumanE more favorably due to its sophisticated mechanics.

Another factor we considered was the size of the test groups. The size of each test group should be sufficiently large to allow for more precision in the analysis of the test results. On the other hand, we did not want the test groups to be overly large as the returns in terms of the accuracy of the test results could be diminishing as the test group size grew.

The evaluation process consisted of the following three tests:

- First test: Test HumanE without learning approach
- Second test: Test HumanE with learning approach without initial policy
- Third test: Test HumanE with learning approach with initial policy

We assigned fifty different testers to each test. We could not repeat the three tests for the same group of testers as they might be influenced by the earlier test data. To obtain consistent feedbacks from the three groups of testers, we gave the testers some guidelines to follow when giving answers. For example during the measurement of the “ease of use” and “performance” metrics, we instructed the testers to give their answers based on the definitions in Table 1 and Table 2.

Scale	**Number of times a tester requests for help or asks questions on the use of HumanE**
Very Bad	>10
Bad	10 – 8
Neutral	7 – 6
Good	5 – 3
Excellent	2 – 0

Table 1. Scale definitions for “ease of use” metric

Scale	Time taken to return matching apartments (sec)
Very Bad	>60
Bad	60 – 30
Neutral	29 – 21
Good	20 – 11
Excellent	10 – 0

Table 2. Scale definitions for "performance" metric

Before the actual evaluation took place, we gave the testers a quick introduction on how to use HumanE. To ensure the objectiveness of the testers' assessments, we chose not to be directly involved throughout the evaluation process (except the scalability test). Instead, a test coordinator with adequate knowledge of HumanE was asked to conduct the experiment.

Typically, the main method to show that a variable affects the outcome is to hold all other variables constant while changing only that variable. Preferably the experiment should be conducted in such a way that the users do not know the state of the variable so that they are unable to help the result even if they want to. Thus to ensure the accuracy of the test results, an identical interface was used to test HumanE 1) without, 2) with the learning approach (excludes initial policy) and 3) with the learning approach (includes initial policy) while the user was not informed of whether HumanE was learning or not.

The objective of each test is to allow the tester to arrive at a satisfactory profile. Each tester was asked to select his or her desired apartments using HumanE's web interface. The test was considered completed when the user declared that he or she was satisfied with the list of desired apartments stored in the "favorites" list. Subsequently, the user was allowed to keep the profile created by printing out a hard copy of the "favorites" list. In each of the three tests, the user was not told whether HumanE was used in helping him or her develop the profile.

The user went through the entire profile creation process without much intervention from the test coordinator. The only assistance that was provided by the test coordinator was to clarify some questions asked by a few users pertaining to navigation and program operation.

At the end of each test, the values of the five metrics were recorded. For the "ease of use" and "performance" metrics, each tester was asked to rate them from a scale of one to five (i.e. 1: Very Bad, 2: Bad, 3: Neutral, 4: Good, 5: Excellent) for the three tests. For each test, the values for the "time taken to create a profile" and "number of profile changes" metrics were recorded and computed by HumanE. Since HumanE recorded the login time and logoff time for each tester, HumanE was able to compute the value for the "time taken to create a profile" metric by subtracting the logoff time from the login time. And because every profile modification was recorded, HumanE was able to provide the value of the "number of profile changes" metric for each tester.

4.3 Experimental Results

The results from the experiments conducted are tabulated in the following figures:

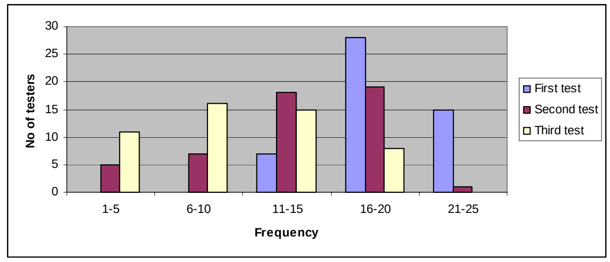

Figure 6. Test result summary for "number of profile changes" metric

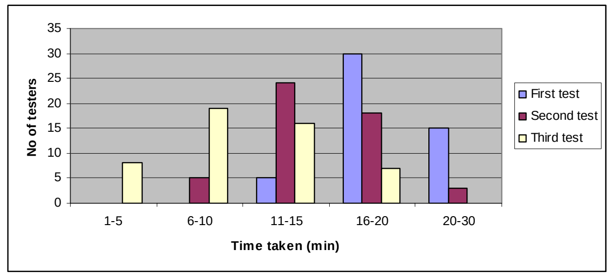

Figure 7. Test result summary for "time taken to create a profile" metric

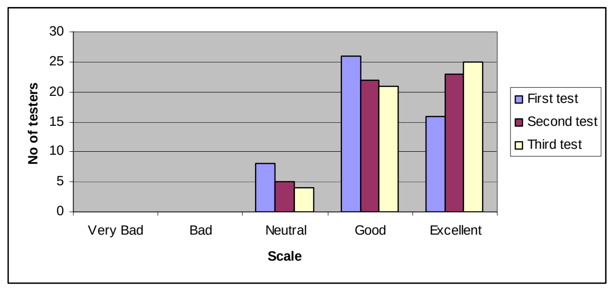

Figure 8. Test result summary for "ease of use" metric

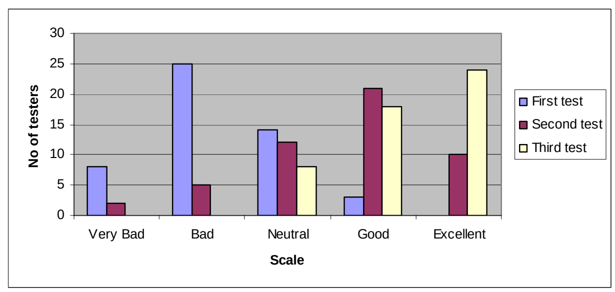

Figure 9. Test result summary for "performance" metric

4.4 Discussion

The test results for the first test for the "number of profile changes" metric showed that most testers took eleven to twenty-five profile changes before converging on a satisfactory profile. The test results for the second test for the "number of profile changes" metric showed some improvement as most testers took eleven to twenty searches. The test results for the third test for the "number of searches" metric showed further improvement as most testers took one to fifteen searches. Thus, it is evident that testers tend to make less number of profile changes with HumanE's assistance and even lesser number when HumanE becomes more intelligent with the supply of the initial policy.

Similarly, the time taken to create a satisfactory profile decreased as we introduced a more intelligent HumanE with each test. The test results for the first test for the "time taken to create a profile" metric showed that most testers took sixteen to thirty minutes while most testers in the second test took less time i.e. from eleven to twenty minutes. However, the testers from the third test took the least time as most of them spent six to fifteen minutes. Thus, it is clear that HumanE can reduce the time taken by users when creating and refining their profiles.

The test results for the three tests for the "ease of use" metric are quite similar indicating that almost all of the testers are happy with using HumanE regardless of whether the learning system with or without the initial policy was present or not. Hence, it is safe to say that using HumanE can result in increased customer satisfaction during the apartment selection process.

The test results for the "performance" metric for the first test apparently showed that majority of the testers were not satisfied with the average quality of the "recommended apartments" shown to them for selection and the average response time taken to display an apartment listing.

Quite a number of them perceived HumanE as a search engine for apartment listings and they were not satisfied with the perceived browsing metaphor which is offered by typical search engines. Even though many testers were fairly happy that they were given complete control over the entire profile creation process, they also voiced out their displeasure of having to make many tedious profile changes before converging on a good profile.

On the other hand, the test results for the second test and the third test showed that the majority of testers preferred to use HumanE to assist them during the apartment selection process. Obviously, the use of HumanE can increase customer satisfaction.

In summary, the experimental results showed that the use of HumanE for complex multidimensional domains such as real estate can result in higher customer satisfaction as it can learn faster via a supplied initial policy and is able to elicit trust from users through its user-friendly interface, quality recommendations and excellent performance.

5. FUTURE WORK

The development of HumanE will continue to evolve particularly in a different domain i.e. vacation plans. In future versions of HumanE, we would like to incorporate some of the following features to improve its usefulness.

- Refine the initial policy refining algorithm based on the results obtained using more sophisticated data mining tools.
- The ability to ask the user questions in natural language, allow the user to enter the response in natural language, and finally understand the response obtained for profile refinement.
- The ability to seek advice from users with similar profiles via email, interpret the reply so as to refine the profile.
- The ability to submit user profile to multiple domain-specific web sites, and show the user the results online. The agent will also need to parse and understand the listing obtained for profile refinement.

6. CONCLUSION

HumanE addresses the problem of poor learning when implementing online implementation of large-scale autonomous agent-based recommender systems for several complex domains through the use of a supplied initial policy which allows it to make more "knowledgeable" exploratory recommendations.

We feel that existing implementations of interactive learning method for online systems are simply impractical as the state-action space is simply too large for the agent to explore within its lifetime. This is further exacerbated by the short attention time-span of typical online users.

It seems easier and more intuitive for the application developer to specify what the agent should be doing and to let it learn the fine details of how to do it. The key strength of our approach is that, by incorporating an initial policy or prior knowledge, HumanE is able to provide better recommendations within a shorter time span.

This is because the initial policy has generated some experiences or knowledge about the real estate domain which HumanE can use throughout the interactive learning process. No longer does the user need to face an agent that does not know anything about the task to be completed. We believe that this approach is far more practical and effective than current implementations [1, 4, 5, 11].

We also postulate, contrary to the experimental results obtained from past research [16], that a good initial policy is critical to the success of HumanE from a reward perspective as the user usually takes less time to build a good profile. Good initial policies head directly for the goal state and they typically do not expose much of the state-space, since their trajectories through it are much more directed. This behavior is actually quite desirable as most online users generally have little patience and want to see a good profile built quickly.

Finally, transferring the work done here to another different domain such as vacation plans, insurance, mutual funds, and mortgages would not require a rocket scientist. The main requirement would be to find a domain expert who would be able to identify the key features of the complex objects in the domain. Creating an initial policy would require the identification of "good" and "bad" features and the classification of features into loosely connected groups.

7. REFERENCES

[1] Brooks, R.A. (1986), A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2:14-23.

[2] Burke, D., Hammond, K. J., and Young, B. C. (1997). "The FindMe Approach to Assisted Browsing," IEEE Expert, pp. 32-40, July-August 1997.

[3] Burke, D., Hammond, K. J., and Young, B. C. (1997). "The FindMe Approach to Assisted Browsing," IEEE Expert, pp. 32-40, July-August 1997.

[4] Franklin, Stan and Graesser, Art: Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.

[5] Decision Support Systems and Intelligent Systems, Efraim Turban and Jay E. Aronson. Copyright 1998, Prentice Hall, Upper Saddle River, NJ.

[6] Evaluating Websites URL: http://trochim.human.cornell.edu/ WebEval/webeval.htm

[7] Gao X., Sterling L. (1998). Classified Advertisement Search Agent (CASA): a knowledge-based information agent for searching semi-structured text. Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, pp. 621-624.

[8] K. Kraemer (ed.), The Information Systems Research Challenge: Survey Research Methods, K. Kraemer (ed.), Boston: Harvard Business School, 1991, pp 299-315.

[9] Newsted, P. R., W. Chin, O. Ngwenyama, and A. Lee, "Resolved: Surveys have Outlived their Usefulness in IS Research," Panel presented at the 1996 International Conference on Information Systems, December 17, 1997 ,Cleveland, Ohio.

[10] Pinsonneault, A. & K. Kraemer, "Survey research methodology in management information systems," Journal of Management Information Systems, Fall, 1993, pp 75-105.

[11] Schneiderman, B., Direct Manipulation: A Step Beyond Programming Languages, IEEE Computer, Vol. 16, No. 8, pp. 57-69, 1983.

[12] Shearin, S. and Lieberman, H. Intelligent Profiling by Example, in Proceedings of Conference on Intelligent User Interfaces, ACM Press, 2001.

[13] Singapore-Real-Estate URL: http://www.singapore-real-estate.com/

[14] SingaporeResidentialProperties URL: http://www.singapore residentialproperties .com/

[15] SurfJet Agent URL: http:// www.leptonicsystems. com/surfjet/

[16] William D. Smart and Leslie Pack Kaelbling, "Effective Reinforcement Learning for Mobile Robots," Proceedings of the IEEE International Conference on Robotics and Automation, 2002.