Value-focused GAI network structure elicitation given a domain Ontology

by

Abeera Farzana Al Baqui

B.A, Gettysburg College, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

October 2007

ABSTRACT

Making optimal decisions is important yet challenging. A decision maker has to take into account her preferences when she needs to make decisions. Often such preferences are over features in the world and exhibit certain structure. It is possible to exploit this structure by making assumptions of independence, to acquire a decision maker's preferences. However, a compromise must be observed while making these assumptions –too many independence assumptions means the preference model acquired is likely to be inaccurate, however too few assumptions yields an overly complex model. A *Generalized Additive Independence (GAI) Network* establishes a good compromise between the accuracy and the generality of the model. A GAI network represents a decision maker's preferences in terms of its structure and a set of utilities. Decision theory provides methods of obtaining both the structure and the utilities of a GAI network; however, these methods are too time consuming, error prone and therefore impractical. Several researchers have investigated methods for simplifying the elicitation procedures for GAI network utilities. However, elicitation of a GAI network structure has not received much attention. *Value Focused Thinking (VFT)* could be a promising solution to this problem, as it can be used to reduce the number of elicitations required to build a DM's GAI network structure as opposed to traditional decision theoretic methods such as standard gambles.

VFT proposes that decision-making should start by decomposing a decision maker's values into additively independent objectives (these are the fundamental objectives). VFT shows how to acquire a decision maker's objectives and map attributes of a domain onto these objectives. We assume that the attributes of a domain are represented in an ontology (which specifies the vocabulary to describe the domain). The decision maker can specify how these attributes fulfill their fundamental objectives.

It is tempting to build a system where non-expert decision makers, minimally trained in concepts of VFT, Ontologies and GAI networks, may express their preferences by 1) specifying objectives and 2) indicating attributes that fulfill these objectives thus creating a value tree. Once a decision maker's value tree is elicited, it is possible to build a corresponding GAI network.

However, it is required that the structure of a decision maker's value tree adhere to the independence assumptions made for GAI networks. We set up an experiment to test whether the structure obtained from eliciting a decision maker's value tree in this manner follows GAI network independence assumptions. We tested this hypothesis in the real-estate domain and

found that the resulting structure does not reflect the independence assumptions of a GAI network. We conclude by discussing implications and suggest changes to our original approach.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to recognize the supervision and advice given by Dr. David Poole and Dr. Giuseppe Carenini during the writing and research for this thesis.

I would also like to acknowledge all the assistance I received with Ruby on Rails and ActiveRDF from Eyal Oren.

DEDICATION

I dedicate this thesis to you, Paul - for his support, criticism and vision. Every conversation, discussion, argument and agreement regarding my work is finally on paper. Thank you for being my beacon for those dark days when light was nowhere to be found.

Chapter 1. INTRODUCTION

Every day we are faced with making decisions. Some decision are simple and trivial such as "what do I wear to work today?" or "what do I make for dinner tonight?", whereas others have major consequences such as "which job offer should I accept?" or "which apartment do I want to live in?" Decisions with high stakes require a decision maker to contemplate, the pros and cons of every action and all manners of trade-offs, to obtain the optimal result. Often, when faced with such decisions, we take into account the state of the world around us, e.g. locations of places to commute and our own preferences, e.g. "I would like a dishwasher in my kitchen". When making decisions we make trade-offs, between our multiple preferences and the state of the world, to attain the most value from taking an action towards achieving that decision.

Our preferences when making decisions are complex and multifaceted. They are expressed in terms of features in the world and properties of individuals (attributes) in the world. For instance, *size of master bedroom* and *facing direction of windows* are different properties for each individual apartment; however, the fact that an apartment has walls and doors are features in the world that are independent of any individual apartment. Theoretically, the assessment of preferences over every combination of values of attributes may be needed to completely elicit our value (utility) for an outcome; but in practice, the large size of the outcome set prevents such a procedure to be feasible. It would not be wise to ask ourselves when regarding an apartment, "do I prefer an apartment with south facing windows or an apartment with south-west facing windows?" and "do I prefer an apartment with south-west facing windows and a dishwasher?" and so on for all possible combinations of features and attributes in apartments. Fortunately, our preferences display an underlying structure [Fishburn. 1970, Keeney. 1976] which may be exploited to make elicitation more feasible. One such structural assumption is that of additive independence between attributes.

A careful trade-off must be made when making independence assumptions about the structure of preferences. If too many independence assumptions are made, the preference model obtained is inaccurate. However, if too few independence assumptions are made, the model is overly complex and the initial problem of easing elicitation remains.

Generalized additive independence (GAI) [Braziunas, et al. 2005] decompositions have recently gained attention because of their flexibility towards the accuracy and the complexity of a decision maker's preference model. GAI is a generalization of the additive model, where independence holds among certain *subsets* of attributes, rather than individual attributes. A GAI network is a graphical model, which represents a GAI decomposition. A GAI network consists of a structure of the preferences and the utilities of attributes that make up the structure.

Elicitation of a decision maker's preferences to build a GAI network using traditional decision theory tools (such as standard gamble queries) is quite impractical. While recent studies [Braziunas, et al. 2005, Gonzales, et al. 2004] have investigated methods of simplifying the utility elicitation of a GAI network, not much work has been done regarding simplifying elicitation of the GAI network structure. We propose value-focused thinking [Keeney. 1996] as an exploratory means to simplify elicitation of the GAI network structure.

Value-focused thinking concentrates on obtaining a decision maker's values. For instance, when deciding which apartment to choose, a decision maker may consider the following objective "Minimize commute time". This general objective can be sub-divided to two or more specific fundamental objectives: "Minimize commute time to work", "Minimize commute time to school" etc.

Value-focused thinking asks a decision maker to consider the objectives behind making a decision before considering the alternatives (outcomes). It allows a decision maker to create a value tree which is a hierarchically organized list of fundamental and means objectives. The creation of a value tree requires a decision maker to state explicitly which and how attributes in the domain fulfill certain objectives. Thus, to fulfill the objective "Minimize commute time to work", a necessary attribute might be *close to bus stop*. One paramount assumption in value-focused thinking is the necessity for a decision maker's value tree to be decomposable into additively independent fundamental objectives.

Keeney has provided an efficient and systematic method of eliciting a decision maker's value tree. We briefly describe this process in Section 2.4. Thus, a promising way of acquiring a GAI network structure may be to use value-focused thinking to acquire a set of fundamental objectives and map sets of attributes under fundamental objectives as additively independent factors.

The vocabulary describing the domain for which a decision is being made contains attributes that also exhibit an inherent hierarchical structure, namely in the form of an ontology. An ontology simplifies a decision maker's tasks in expressing her value tree in terms of attributes. A systematic breakdown of the domain into its sub-parts and their properties allows the decision maker to consider all aspects of the domain without overlooking any detail.

It is therefore quite tempting to build a system for non-expert decision makers minimally trained in the concepts of value-focused thinking and ontologies to express their preferences when making a decision. We may envision that such a system would allow a decision maker to specify her value trees in terms of her fundamental objectives; she would also indicate the attributes from a domain ontology that fulfill her fundamental objectives. A decision maker's value tree obtained in such a manner could be used to build the structure of a GAI network.

However, for such a system to be built we need to ensure that the structure obtained from a decision maker's value tree adheres to all the independence assumptions inherent in the resulting GAI network. We built a system and set up an experiment in the real-estate domain to test the following hypotheses:

- On preference Elicitation: It is possible to elicit the structure of a sound GAI network by having a minimally trained DM 1) create a value tree by following Value Focused Thinking and 2) map the domain attributes from an ontology into the fundamental objectives of the value tree (that are fulfilled) by using the interface developed.
- On verification of the output of the elicitation procedure: It is possible to measure the quality of the obtained GAI network by asking a (minimally trained) DM to answer comparison gamble questions comparing two lotteries.

This thesis is structured in the following manner. Chapter 2 discusses the history and background information for decision theory, preference structures and GAI networks, value-focused thinking and ontologies. Chapter 3 delves into our contribution and the details of our approach. Chapter 4 deals with the test-bed system built to test our hypotheses. Chapter 5 discusses the experimental procedure setup to test the above hypotheses and discusses our results. Finally, Chapter 6 wraps up this thesis by providing some suggestions for future work and the conclusion of our research.

Chapter 2. BACKGROUND

2.1. HISTORY

Complex decision problems with multiple objectives have been faced since the beginning of civilization. The history of decision analysis, however, is not very long. In 1738, Daniel Bernoulli used the concept of utility to explain the St. Petersburg paradox [Bernoulli. 1954]. This insight created the foundations of utility theory, a numerical measure describing the value of alternatives; and the notation of a utility function, the numerical measure itself. In the next century or so, utility theory was used to explain economic behavior. Some utilitarian philosophers used utility theory as a tool for constructing the theory of ethics. The theory did not gain much popularity, however, as it was not yet possible to measure a person's utility function.

In the 1940s and 1950s, utility theory was placed on sound theoretical foundation. *Game theory* was developed to describe the behavior of rational people when engaging with others with conflicting goals. In 1944, John von Neumann and Oskar Morgenstern published *The Theory of Games and Economic Behavior*, which to this day counts as the most influential contribution in decision theory. A wave of new researchers and practitioners, from statistics and operations research community, ignited by this new theory of utility, developed approaches and tools to help Decision Makers (DMs) in difficult decisions. In contemporary times, decision theorists use artificial intelligence (AI) to apply and extend techniques developed in decision science (how ideal decisions should be made and optimal decisions can be reached) for addressing decision making under uncertainty, as reasoning about actions under incomplete information and scarce resources is central to solving difficult problems in AI.

2.2. DECISION THEORY

Decision theory lies at the intersection of many academic disciplines – statistics, economics, psychology, game theory, operations research, and others. Assuming a set of axioms for rational behavior, it provides a theory for modeling user preferences and making optimal decisions based on these preferences. The following summary of main concepts is based on Fishburn. 1970, Keeney. 1976, von Neumann, et al. 1944.

***Definition 1.* A decision maker (DM) is a person, organization or any other decision-making entity, empowered to make decisions concerning the problem at hand.**

Definition 2. **A decision is an irrevocable allocation of resources under control of the decision maker.**

Definition 3. **Preferences describe a decision maker's relative valuations for possible states of the world, or outcomes.**

The valuation of an outcome may be based on the traditional attributes of money and time, as well as on other dimensions of value including pleasure, pain, life years and computational effort.*

A DM has to select a single alternative (or action) $a \in A$ from the set of available alternatives. Assume a set of attributes $X_1, X_2, \ldots, X_n$, each with finite domains. These define a set of *outcomes* $X = X_1 \times \ldots \times X_n$ over which a DM has preferences. An *outcome,* $x \in X$ of the chosen action depends on the state of the world $\theta \in \Theta$. The consequence function $c : A \times \Theta \mapsto X$ maps each action and world state into an outcome. DM preferences can be expressed by a value, or *utility*, function $v : X \mapsto \mathbb{R}$ that measures desirability of outcomes. The goal is to select an action $a \in A$ that leads to best outcomes. If the world state θ is known, the set of outcomes is equivalent to the set of alternatives. When uncertainty over world states is quantified probabilistically, *utility theory* prescribes an action that leads to the highest expected utility (EU).

The outcome space itself is defined in multi-dimensions. Most interesting problems fall in this category, and we explore one way of exploiting the structure in multidimensional outcome spaces to acquire a decision maker's preferences.

A DM's preference for one object over another can be represented by one of the following three binary relations: strict preference, weak preference and indifference.

- $a \succ b$ means that the DM *strictly* prefers the object a to the object b. In other words, if a choice between a and b was offered to the DM, she would be disappointed if she had to select the object b.
- $a \succsim b$ means that the DM *weakly* prefers the object a to the object b. That is, according to the DM the object a is at least as good as the object b. If the DM was offered a choice between the object a and b, she would not be disappointed if she was forced to take the object a.

---

* Definitions from [Helsinki University of Technology (Systems Analysis Laboratory). 2005]

- $a \sim b$ means that the DM is *indifferent* between the object a and b. In other words, if she was offered a choice between a and b, she would not be disappointed if she was subsequently forced to take either of the options.

Decision theory is based on the axioms of probability and utility, as mentioned above. Where probability theory provides a framework for coherent assignment of beliefs with incomplete information, utility theory introduces a set of principles for consistency among *preferences* and *decisions*.

Since we are concerned with a DM's preferences under uncertainty, we will focus more on utility theory rather than preference theory. Utility theory is based on a set of simple axioms or rules concerning choices under uncertainty, these axioms are discussed below [Fishburn. 1981].

2.2.1. AXIOMS OF UTILITY THEORY

The first set of axioms concerns *preferences for outcomes under certainty*.

- The axiom of *orderability* asserts that all outcomes are comparable, even if described by many attributes. Thus, for any two possible outcomes *x* and *y*, either one prefers *x* to *y* or one prefers *y* to *x*, or one is indifferent between them.
- The axiom of *transitivity* asserts that these orderings are consistent; that is, if one prefers *x* to *y* and *y* to *z*, then one prefers *x* to *z*.

These axioms, together with two auxiliary axioms, ensure a weak preference ordering of all outcomes. This result implies the existence of a scalar value function *V(x)*, which maps from all outcomes *x* into a scalar "value" such that one will always prefer outcomes with a higher "value".

The second set of axioms describes *preferences under uncertainty*. They involve the notion of a lottery, an uncertain situation with more than one possible outcome. Each outcome has an assignable probability of occurrence.

Definition 4. **Lottery: Let x_i be an outcome and let X be a set of outcomes. Let P be a simple probability measure on X, thus $P = (p(x_1), p(x_2), \dots, p(x_n))$ where in a lottery $p(x_i)$ are probabilities of outcome $x_i \in X$ occurring, i.e. $p(x_i) \geq 0$ for all i = 1, ..., n and $\sum_{i=1}^{n} p(x_i) = 1$**

- The axiom of *monotonicity* says that, when comparing two lotteries, each with the same two alternative outcomes but different probabilities, a DM should prefer the lottery that has the higher probability of the preferred outcome.

- The axiom of *decomposability* says that a DM should be indifferent between lotteries that have the same set of eventual outcomes and probabilities, even if they are reached by different means. For example, a lottery whose outcomes are other lotteries can be decomposed into an equivalent one-stage lottery using the standard rules of probability.
- The axiom of *substitutability* asserts that, if a DM is indifferent between a lottery and some certain outcome (the certainty equivalent of the lottery), then substituting one for the other as a possible outcome in some more complex lottery should not affect her preference for that lottery.
- The axiom of *continuity* says that, if one prefers outcome x to y, and y to z, then there is some probability p such that one is indifferent between getting the intermediate outcome y for sure and a lottery with a p chance of x (the best outcome) and $(1 - p)$ chance of z (the worst outcome).

It follows from accepting the axioms of utility that there exists a scalar utility function $U(x)$, which assigns a number on a cardinal scale to each outcome x, indicating its relative desirability. Further, it follows that when there is uncertainty about x, preferred decisions d are those that maximize the expected utility $E[U(x;\ d)|\xi]$ over the probability distribution for x.

The consistency criteria embodied in classical decision theory can be stated as follows [Horvitz. 1988]:

> *Given a set of preferences expressed as a utility function, beliefs expressed as probability distributions, and a set of decision alternatives, a decision maker should choose that course of action that maximizes expected utility (EU).*

The power of this result is that it allows preferences for distributions of outcomes to be computed from preferences expressed for the individual outcomes. Thus, it may be used as a tool to help people think about complex choices by decomposing them into simpler choices.

2.2.2. UTILITY FUNCTION

***Definition* 5. A utility function $u: X \longmapsto \mathbb{R}$ rationalizes a preference relation $\precsim$ on X if for every $x_i, x_j \in X, u(x_i) \leq u(x_j)$ if and only if $x_i \precsim x_j$.**

If u rationalizes $\precsim$, then this implies $\precsim$ is complete and transitive, and hence rational. Utility functions enable a compact representation of preferences over outcomes. For describing preferences over n outcomes, n real numbers are required; the outcome with a greater utility is preferred to outcomes with smaller utilities. To measure the utility of each outcome in X we can perform standard gamble queries.

2.2.3. Standard Gamble Query

A standard gamble is a paired comparison between two alternatives where one alternative is a lottery between two possible outcomes and the other alternative is a sure outcome. Consider the standard gamble defined in Figure 1, where lottery L yields a p chance of consequence $x^{\top}$ (the best outcome) and a 1-p chance $x^{\perp}$ (the worst outcome) and x, which is a sure outcome. A decision maker's utility for outcome x can be obtained by measuring the probability p at which the decision maker is indifferent between lottery L and the outcome x.

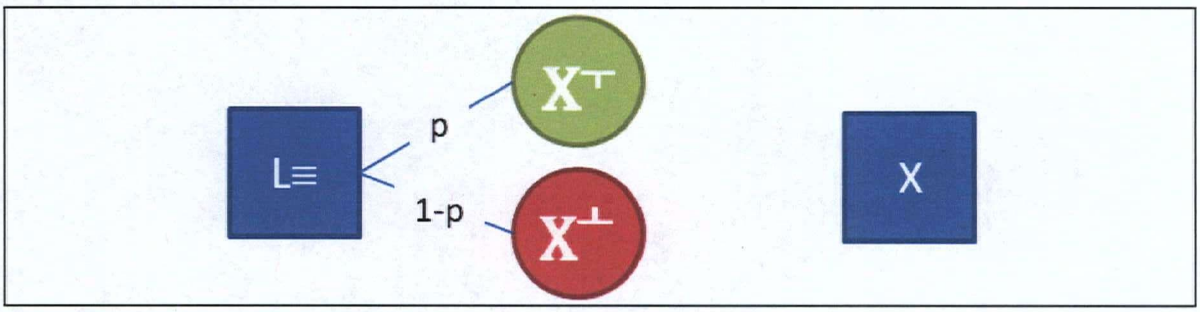


Figure 1: Illustration for a Standard Gamble

2.3. Structure of Preferences

An attribute X_1 has a set of possible values $x_1^1, x_1^2 \dots x_1^n$. An outcome is defined by the values of each attribute in the attribute set $\{X_1, X_2, \dots X_3\}$. Since the number of outcomes is exponential in the number of attributes, specifying the utility of each outcome is infeasible in most practical applications. The assessment of preferences for every outcome compared to the best and the worst outcome is needed to elicit completely the DM's utility. The large size of the outcome set makes such a procedure impractical. Fortunately, preferences often have an underlying structure that can be exploited to reduce the elicitation burden. Several structures described in terms of different independence concepts have emerged from the multi-attribute utility theory community the most popular of which is the additive decomposition.

A DM's preferences can be expressed concisely if it exhibits sufficient structure such as an additive decomposition. Additive independence is one structural assumption commonly used in practice.

2.3.1. ADDITIVE INDEPENDENCE

Definition 6. **Additive Independence: Let L_1 and L_2 be any pair of lotteries and let p and q be their respective probability distributions over the outcome set. Then $X_1, \ldots, X_n$ are additively independent for $\succsim$ if p and q having the same marginals on every X_i implies that both lotteries are indifferent, i.e. $L_1 \succsim L_2$ and $L_2 \succsim L_1$ (or $L_1 \sim L_2$ for short).**

[Bacchus, et al. 1995] illustrates additive independence on the following example:

$$\text{Let } X = X_1 \times X_2 \textit{ where } X_1 = \{a_1, b_1\} \textit{ and } X_2 = \{a_2, b_2\}.$$

Let L_1 and L_2 be lotteries whose respective probability distributions on X are p and q.

Assume

$$p(a_1, a_2) = p(a_1, b_2) = p(b_1, a_2) = p(b_1, b_2) = 1/4$$

$$q(a_1, a_2) = q(b_1, b_2) = 1/2 \text{ and}$$

$$q(a_1, b_2) = q(b_1, a_2) = 0.$$

Then p and q have the same marginals on X_1 and X_2, since

$$p(a_1) = q(a_1) = 1/2,\ p(b_1) = q(b_1) = 1/2,$$

$$p(a_2) = q(a_2) = 1/2, p(b_2) = q(b_2) = 1/2.$$

So, under additive independence, lotteries L_1 and L_2 should be indifferent. Given a complete utility function that can be derived using standard gambles, we can test dependence between attributes. However, comparison gambles make this task more feasible and direct.

2.3.2. COMPARISON GAMBLE QUERY

A comparison gamble is a decision aid tool to adequately measure the dependence between two attributes X_1 and X_2. A comparison gamble is a paired comparison between two alternatives where each alternative is a lottery between two possible outcomes. Consider two lotteries L_1 and L_2 as defined in Figure 2. Lottery L_1 yields a p chance at the outcome $(x^\top, y^\top)$ and a 1-p chance at the outcome $(x^\perp, y^\perp)$ and lottery L_2 yields equal 0.5 chances at each of $(x^\top, y^\perp)$ and $(x^\perp, y^\top)$*. In the case when p equals 0.5, note that both lotteries have an equal (i.e. 0.5) chance at either $x^\top$ or $x^\perp$ and also that both have an equal 0.5 chance at either $y^\top$ or $y^\perp$. By definition then the marginal probability distributions on each of the attributes of X and Y are the same in

---

* $x^\top, x^\perp, y^\top$ and $y^\perp$ are used here as possible values for attributes. For attributes X and Y to be additively independent, this needs to hold for any pair of values for the attributes.

both the lotteries when p equals 0.5. Thus, if X and Y are additively independent the DM must be indifferent between lotteries L_1 and L_2. This same indifference condition must hold if either or both of $x^{\perp}$ and $y^{\perp}$ are changed, because L_1 and L_2 will still have the same marginal probability distributions on the two attributes.

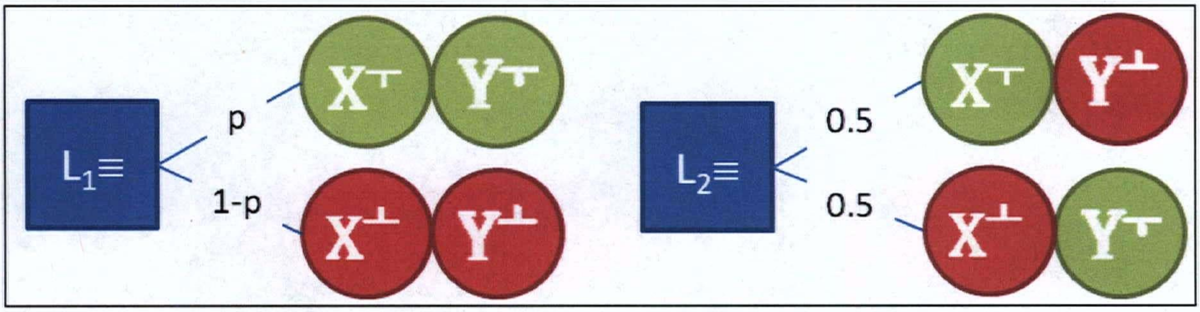


Figure 2: Illustration for a Combination Gamble

Attributes X and Y are considered dependent when a DM is indifferent between lotteries L_1 and L_2 when $p \neq 0.5$.

2.3.3. Attribute Dependence

To measure the degree of dependence between two attributes, the probability *p* at which a DM is indifferent between lotteries L_1 and L_2 is measured. For example, for the comparison gamble shown above, when a DM has the following preference $L_1 \succ L_2$, decreasing the probability *p* and asking the DM the same comparison gamble query might result in a preference of $L_1 \sim L_2$, thus the degree of dependence between the two attributes X and Y would corresponds to the new *p* value.

2.3.4. Substitutes and Complements

Attributes interact with each other in many ways. One interaction that attributes may have can be described as being *substitutes* of each other, i.e. one attribute's presence replaces the absence of another attribute in terms of value. For example, the attributes *close to park* and *close to beach* may behave as substitutes for a DM. Conversely, attributes may complement each other, i.e. the presence of both attributes is required for the DM to be satisfied. For example, the attributes *facing South* and *has window* are *complementary* attributes, as having only both would satisfy the DM's need for adequate lighting.

This categorization of the interaction between attributes can be measured using a comparison gamble query. If a DM's preference is $L_1 \sim L_2$ when $p > 0.5$ for the comparison gamble shown above, this indicates that the attributes are substitutes, i.e. a DM would rather prefer a gamble where she has one or the other attribute. Correspondingly, when $p < 0.5$ and a DM's preference is $L_1 \sim L_2$ the attributes are complementary, i.e. the DM would rather gamble to obtain both the attributes then end up with one or the other attribute.

2.3.5. Implications of Additive Independence

Under a strong independence assumption—specifically, that the DM is indifferent among lotteries that have same marginals on each attribute—u can be written as a sum of single-attribute sub-utility functions:

$$u(\boldsymbol{x}) = \sum_{i=1}^{n} u_i(x_i) = \sum_{i=1}^{n} \lambda_i v_i(x_i)$$

This simple factorization exploits subutility functions $u_i(x_i) = \lambda_i v_i(x_i)$, which themselves depend on local value functions v_i and scaling constants λ_i. The assumed utility independence among attributes allows elicitation to proceed locally: specifically, the v_i can be elicited independently of other attribute values. Since each attribute is utility independent, each attribute's best and worst levels can be determined separately. Since utility functions corresponding to $\succsim$ are unique only up to positive affine transformation, it is customary to set the utility of the best value $(x_i^{\top})$ for the attribute to 1, and the utility of the worst value $(x_i^{\perp})$ for the attribute to 0.

A standard gamble query between x_i^k and a lottery between the two anchor outcomes, the best and worst levels of attribute x_i: $\langle p, x_i^{\top}; 1 - p, x_i^{\perp} \rangle$ is put to the DM. Eliciting the indifference for a specific p for this gamble will return the v_i for the attribute as:

$$u(x_i^k) = pu(x_i^{\top}) + (1 - p)u(x_i^{\perp})$$

and therefore, because of the additive form u,

$$v(x_i^k) = pv_i(x_i^{\top}) + (1 - p)v_i(x_i^{\perp})$$

Thus, if a DM is indifferent for the given standard gamble, then $u(x_i^k) = p$. To elicit the global scaling constants, queries involving full outcomes are asked [Braziunas, et al. 2005].

Additive Independence simplifies the elicitation procedures yet compels the DM's preferences to satisfy very stringent constraints. This form of independence captures the fact that preferences only depend on the marginal probabilities on each attribute, but rules out interactions between attributes, which are quite common in DMs preferences [Bacchus, et al. 1995], therefore such functional form cannot be applied in many practical situations. One-way out is to allow some limited forms of interactions between attributes; for instance, by separating the utility function into a sum of sub-utilities on sets of interacting attributes: this leads to generalized additive independence (GAI) decompositions.

2.3.6. GENERALIZED ADDITIVE INDEPENDENCE

Definition 7. **Generalized Additive Independence: Let L_1 and L_2 lotteries and let p and q be their probability distributions over the outcome set. Let $Z_1, \ldots, Z_k$ be some possibly overlapping subsets of $N=\{1, \ldots, n\}$ such that $N = \cup_{i=1}^{k} Z_i$ and let $X_{Z_i} = \{X_j : j \in Z_i\}$. Then $X_{Z_1}, \ldots, X_{Z_k}$ are generalized additively independent for $\succsim$ if the equality of the marginals of p and q on all X_{Z_i}'s implies that $L_1 \sim L_2$. [Gonzales, et al. 2004]**

The following functional form of the utility called a GAI decomposition can be derived from generalized additive independence:

Let $Z_1, \ldots, Z_k$ be some subsets of $N = \{1, \ldots, n\}$ such that $N = \cup_{i=1}^{k} Z_i$ and $X_{Z_i} = \{X_j : j \in Z_i\}$ are generalized additively independent (GAI) for $\succsim$ iff there exist some real functions $u_i : X_{Z_1} \times \ldots \times X_{Z_i} \mapsto \mathbb{R}$ such that

$$u(x) = \sum_{i=1}^{k} u_i(x_{Z_i}), for\ all\ x = (x_1, \ldots, x_n) \in X$$

where X_{Z_i} denotes the tuple of components of x having their index in Z_i.

GAI decompositions allow great flexibility because they make less stringent assumptions on the kind of relations between attributes. Furthermore, the amount of questions required by the elicitation is manageable thanks to the GAI decomposition itself.

A graphical structure called a GAI network can effectively represent GAI decompositions.

2.3.7. GAI NETWORK

Definition 8. **Generalized Additive Independence network (GAI network): Let $Z_1, \ldots, Z_k$ be some subsets of $N = \{1, \ldots, n\}$ such that $N = \cup_{i=1}^{k} Z_i$. Assume that $\succsim$ is representable by a GAI-decomposable utility $u(x) = \sum_{i=1}^{k} u_i(x_{Z_i}), for\ all\ x \in X$. Then a GAI network representing $u(\cdot)$ is an undirected graph G = (V,E), satisfying the**

following properties:

1. V = $\{X_{Z_1}, \ldots, X_{Z_k}\}$;

2. $(X_{Z_i}, X_{Z_j}) \in E$ iff $\exists\, Z_i \cap Z_j \neq \emptyset$.

Nodes of V are called factors. Every edge $(X_{Z_i}, X_{Z_j}) \in$ E is labeled by $X_{T_{ij}} = X_{Z_i \cap Z_j}$, which is called a separator.

The GAI network provides an additive decomposition of utility functions in situations where single attributes are not additively independent, but (possibly non-disjoint) subsets of attributes are [Fishburn. 1970]. The form of a GAI network is as follows.

Assume a collection $\{Z_1, \ldots Z_k\}$ of (possibly intersecting) factors such that $\bigcup_i Z_i = \{1, \ldots, n\}$ and local subutility functions u_i over X_{Z_i}. Then u is decomposed as:

$$u(X) = u_1(X_{Z_1}) + \cdots + u_k(X_{Z_k})$$

For any GAI decomposition, Definition 8 is explicit as to which factors should be created: these are simply the sets of variables of each sub-utility. For instance, if $u(x_1, \ldots, x_7) = u_1(x_1, x_2, x_3) + u_2(x_3, x_4) + u_3(x_4, x_5) + u_4(x_6, x_7)$ then, as shown in Figure 3.a, factors are $\{X_1, X_2, X_3\}$, $\{X_3, X_4\}$, $\{X_4, X_5\}$ and $\{X_6, X_7\}$.

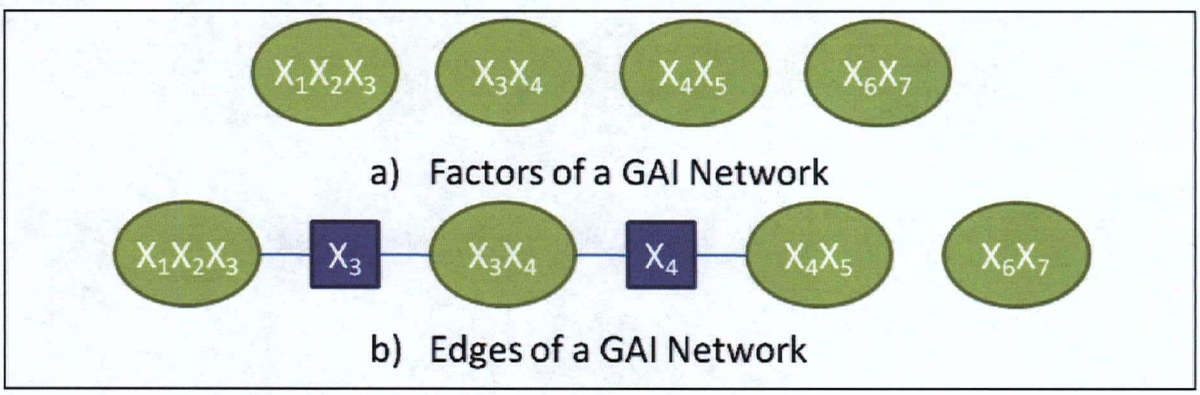


Figure 3: Construction of a GAI net

Property 2 of Definition 8 gives us a clue for determining the set of edges of a GAI network. The following section describes the algorithm to construct a GAI network.

2.3.8. ALGORITHM FOR CONSTRUCTING A GAI NETWORK

The algorithm to construct the GAI network structure, given the local sub-utility functions, is shown below:

```
construct set V = {X_{Z_1}, ..., X_{Z_k}};
for i ∈ {1 . . . , k - 1} do
  for j ∈ {i + 1. . . , k} do
    if Zi ∩ Zj ≠ ∅ then
      add edge (X_{Z_i}, X_{Z_j}) to E
    end
  done
done
```

Applying this algorithm on set $V= \{\{X_1,X_2,X_3\}, \{X_3,X_4\}, \{X_4,X_5\}, \{X_6,X_7\}\}$, sets $\{X_1,X_2,X_3\}$ and $\{X_3,X_4\}$ having a nonempty intersection, an edge would be created between these two factors. Similarly, edge $\{\{X_3,X_4\}, \{X_4,X_5\}\}$ would also be added as X_4 belongs to both factors. Since the intersection between $\{X_6,X_7\}$ and all the other sets in V returns a null set, there is no edge created. Consequently the network in Figure 3.b is a GAI network, representing $u(x_1, \ldots, x_7) = u_1(x_1, x_2, x_3) + u_2(x_3, x_4) + u_3(x_4, x_5) + u_3(x_6, x_7)$.

Eliciting each local sub-utility function has been investigated by [Braziunas, et al. 2005]; however, simplifying the elicitation of the structure of the decomposition has not received much attention.

2.3.9. Structural Elicitation of GAI Networks

Using traditional decision theoretic tools such as comparison gambles for pairs of attributes to determine the set of sub-utility functions to create the GAI network structure is infeasible and time-consuming. The number of comparison gamble queries would be exponential in the number of attributes, also the same comparison gamble query would be asked several times to elicit the accurate probability for measuring the degree of dependence between attributes. We propose value-focused thinking to be a viable approach for this problem as it decomposes a DM's values into additively independent objectives.

2.4. Value Focused Thinking

Value-focused thinking advocates thinking about values first and then evaluating the alternatives present that achieve those values [Keeney. 1996]. Keeney refers to *value-focused thinking* as an iterative approach between expressing values and choosing alternatives; however, the principle is "values first". Keeney details the multiple uses of value-focused thinking, such as uncovering hidden objectives, guiding information collection, interconnecting decisions, guiding strategic thinking, evaluating alternatives etc. Value focused thinking provides a structured way to think about decisions and develop and support subjective judgments that are critical for good decisions.

2.4.1. Objectives

***Definition 9.* Objectives are statements of something that one desires to achieve.**

Generally, objectives are characterized by three features:

- Decision context: the activity being contemplated
- Object: a goal this objective would achieve
- Direction of preferences

For example, in selecting an apartment one of the objectives maybe to **MINIMIZE COMMUTE TIME**. For this objective, the *decision context* is apartment selection, the *object* is the amount of time taken to get from home to workplace and less time taken is *preferable* to more time taken.

- Objective specification does not require the identification of a measure (for example commute time in hours per day) to indicate the level to which the objective is achieved.
- An objective does not quantify the relative desirability of different levels of the object.
- Objectives can be divided into two classes: Fundamental objectives and means objectives.

***Definition 10.* Fundamental objectives characterize an essential reason for interest in the decision situation.**

***Definition 11.* Means objectives are of interest in a decision context because they are a means to achieving fundamental objectives.**

For example, lower rent may appear to be an important objective, but it may be seen important only because it would allow a DM to improve her living standard, to pursue activities that represent fundamental interests. Thus, *low rent* could be seen as a means objective and *increase living standard* as a fundamental objective.

2.4.2. ATTRIBUTES

An objective is measured in terms of a set of attributes. For instance, $X_i(a) = x_i$ indicates the level x_i, where $x_i \in \{x_i^1, \dots x_i^n\}$, to which an objective is achieved in the outcome *a* for the attribute X_i. For example, *Availability of Fire alarm* is one measure for the objective **MAXIMIZE SAFETY**. If an apartment has a fire alarm, then the level at which this objective is achieved for this attribute is 1 and an absence of a fire alarm indicates that the value of this attributes for achieving this objective is 0.

2.4.3. VALUE FOCUSED THINKING PROCESS

Using value-focused thinking, a DM would have to come up with her fundamental objectives and concern herself with which attributes fulfill those objectives, thus reducing the complexity to a polynomial function, i.e. **number of FO in value tree × number of A in the domain.**

During the elicitation of preferences, the DM specifies objectives and attributes that fulfill these objectives as a value tree. As shown in Figure 4*, decision-making using value-focused thinking involves four main phases.

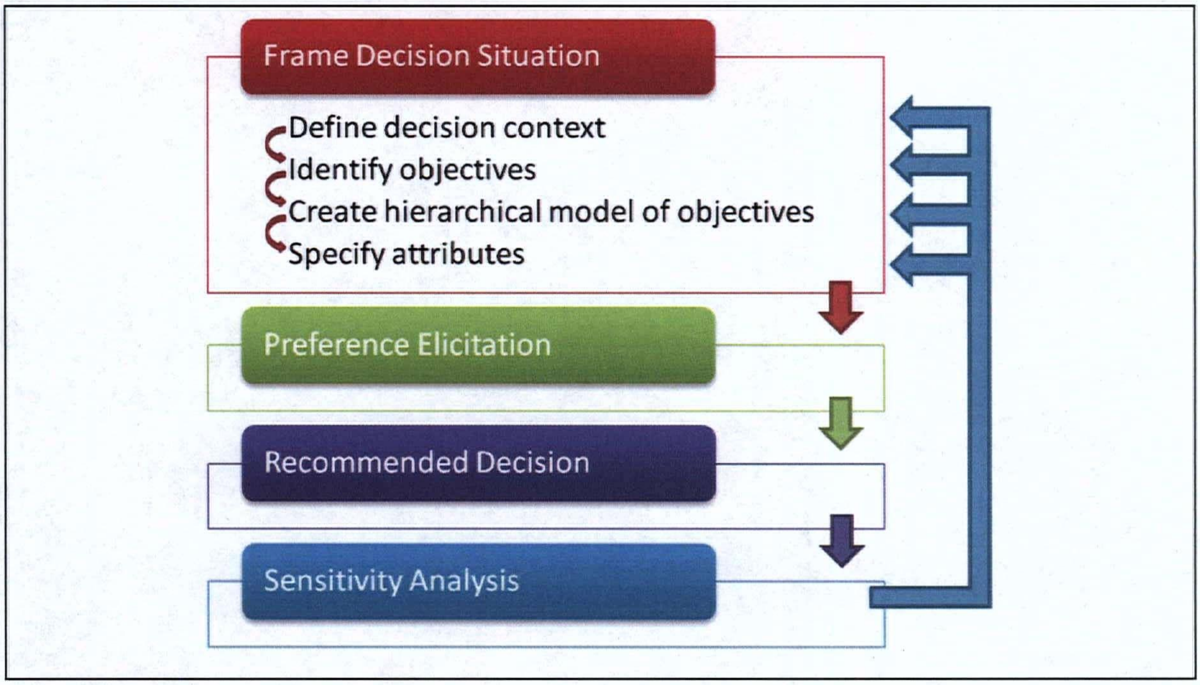


Figure 4: Phases of decision-making using value-focused thinking

---

* Adapted from Helsinki University of Technology [Systems Analysis Laboratory],

For our purposes, we will only concentrate on the first phase of the process as a plethora of research has already been done for each of the other phases [Fishburn. 1970, Braziunas, et al. 2005, Bacchus, et al. 1995]. The process is iterative and quite often long and time consuming. In addition, the DM's preferences and objectives may change and develop during the process.

The main purpose of the *framing a decision situation* phase is to create a better understanding of the decision problem. For example, answers to the following questions should become evident.

- What is important, and relevant?
- What are the objectives?
- What information is available?

Define decision context: by doing this and establishing the nature of the decision problem carefully, the treatment of the real problem can be ensured. A careful specification of the decision context is particularly relevant if several DMs or stakeholders are involved in the decision analysis process. Without a mutual agreement on the decision context, problems are likely to occur in the subsequent phases.

Identifying objectives requires significant creativity. The most obvious way to identify objectives is to ask a decision maker to recapitulate the decision context, and then provide a written list of objectives. Several devices can be used to stimulate the identification.

Creating a hierarchical structure of the objectives, often called a value tree, is to produce a deeper and more accurate and analytic understanding of the problem and a basis for quantitative analysis.

Specification of attributes aims to measure the achievement of objectives in different decision alternatives.

After the decision situation and real nature of the problem is established, objectives are identified. Relations between multiple objectives are analyzed and a hierarchical model is obtained. With a hierarchical model, relations between objectives are more easily understood. The model also creates a basis for further analysis. Attributes measure the extent to which different decision alternatives satisfy the stated objectives. Specification of attributes thus enables the comparison of the alternatives.

The aim of the *preference elicitation* is to measure and estimate the DM's utilities over a set of objectives.

- Measuring preferences is not straightforward. It may be that the DM is not sure about her preferences, she is unable to state them or she is even unaware of them. Furthermore, the DM may act inconsistently and give conflicting statements about her preferences.
- In most cases, the preference elicitation is an iterative process used to ensure the best possible estimates of the DM's preferences.
- Knowing the DM's preferences, information about the attribute levels for different decision alternatives and the DM's value tree can be used to find the most preferred alternative given as a *recommended decision*.

The aim of the *sensitivity analysis* is to explore how changes in the model influence the decision recommendation.

- If a small change in one or several aspects of the value tree causes the recommended decision to change, the decision is said to be sensitive to those changes.
- Recognizing the aspects to which the decision is sensitive enables the DM to concentrate on, or possibly reconsider the issues, which may cause changes in the decision.
- Any part of the decision analysis process, from the identification of the decision problem to the evaluation of the preferences, can be subjected to the sensitivity analysis.

As Figure 4 shows, after the sensitivity analysis the DM may return to earlier phases of the decision analysis process; new objectives may be identified, the value tree may be changed etc. Thus, sensitivity analysis is a central part of the decision analysis cycle.

A value tree, mentioned above, is a hierarchical list of objectives. The degree to which objectives are achieved in different decision alternatives is measured with attributes. Often these attributes are properties of entities that exist in some domain. These entities and attributes can typically be ordered hierarchically as an ontology.

2.5. Ontologies

Attributes that measure the achievement of objectives are specified in a domain ontology.

***Definition 12.* Ontology: A description of the concepts and relationships that can exist.**

Ontologies typically specify the meanings and hierarchical relationships among terms and concepts in a domain. An ontology is made up of several elements:

- Individuals: the basic or "ground level" objects
- Classes: sets, collections, or types of objects
- Properties: features, characteristics, or parameters that objects can have
- Relations: ways that objects can be related to one another

***Definition 13.* An ontology language is a formal language used to encode an ontology.**

There are a number of ontology languages such as the *Web Ontology Language* (OWL). An OWL ontology may include descriptions of classes, along with their related properties and instances. OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. It facilitates greater machine interpretability of content by providing additional vocabulary along with a formal semantics.

OWL currently has three sub-languages (also referred to as 'species'): OWL Lite, OWL DL, and OWL Full. These three increasingly expressive sublanguages are designed for use by specific communities of implementers and DMs.

2.5.1. Domain and Upper Ontologies

A *domain ontology* (or domain-specific ontology) models a specific domain, or part of the world. It represents the particular meanings of terms as they apply to that domain. For example the word floor has many different meanings. An ontology about the domain of architecture would model the "lower horizontal surface of a room" meaning of the word, while an ontology about the domain of mathematical computation would model the "floor function" meaning.

An *upper ontology* (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It contains a core glossary in whose terms objects in a set of domains can be described. There are several standardized/formal upper ontologies available for use, including Dublin Core, GFO, OpenCyc/ResearchCyc, SUMO, and DOLCEl.

Chapter 3. ELICITING A GAI NETWORK BASED ON VALUE FOCUSED THINKING

3.1. PREDICTIONS FROM THEORY

In our approach, we aim to elicit the factors of the GAI network for a DM by asking her to describe her value tree in terms of fundamental objectives and attributes that fulfill these objectives. Attributes bundled under one objective represents attributes that make up a factor of the DM's GAI network.

So, given the DM's value tree, a preference model is generated as a GAI network. We hypothesize that our approach is an effective means for eliciting a DM's GAI network and would complement the local utility elicitation of GAI networks [Braziunas, et al. 2005] for obtaining a DM's preferences.

Since our approach relies on the concepts of value-focused thinking and the independence assumptions made when building a value tree and its corresponding GAI network, a discussion of the predictions made in the theory regarding these concepts is necessary.

According to Keeney, the fundamental objectives in a value-tree should be additively independent. If a user defines their preferences using the simplest set of fundamental objectives, the theory for additive independence predicts that a GAI network built based on a value tree would exhibit these properties:

- Attributes in disconnected factors are independent
- Attributes in the same factor are dependent
- As the number of edges between attributes in factors increases, the dependence between these attributes decreases (weak)

For instance, if a GAI network as shown in Figure 3 (page 13) is built based on a value-tree, the decomposition implies 1) X_6 is additively independent from X_1, X_2, X_3, X_4, X_5, 2) X_1 is additively dependent to X_2, X_3, X_4, X_5, 3) The dependence between X_2 and X_3 is higher than the dependence between X_2 and X_4 and so on.

To test these predictions, we can ask comparison gamble queries to test for independence and degree of dependence if any.

For instance, a comparison gamble query between X_1 and X_6 can be structured as shown in

Figure 5 where $X^{\top}$ represents the best value for attribute X and $X^{\perp}$ represents the worst value. Given a comparison gamble between lotteries L_1 and L_2, the attributes X_1 and X_6 are additively independent when the DM is indifferent between L_1 and L_2 when $p = 0.5$. Similarly, attributes X_1 and X_6 would be dependent if a DM is indifferent between lotteries L_1 and L_2 when $p \neq 0.5$.

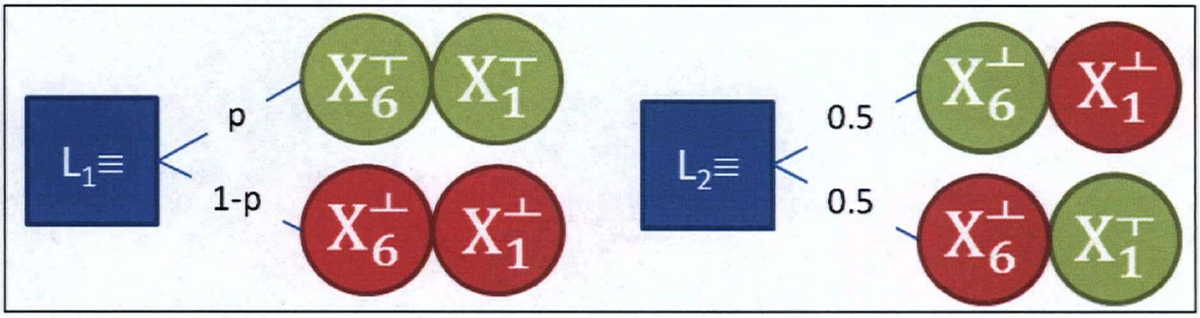


Figure 5: Comparison Gamble for testing independence between attributes X_1 and X_6

To measure the degree of dependence between two attributes, we can elicit the value marked by p in Figure 5, which represents the probability at which a DM is indifferent between lotteries L_1 and L_2. For example, for the comparison gamble shown above, when a DM has the following preference $L_1 \succ L_2$ at $p = 0.5$, we can decrease the probability at p by half to 0.25 and ask the DM the same comparison gamble query. If the DM's preference is now $L_1 \sim L_2$, we can estimate the dependency between attributes X_1 and X_6 to be 0.25. This preference ordering demonstrates two key concepts:

- Attributes X_1 and X_6 are dependent, therefore the factors could either be $\{X_1, X_2, X_3, X_6\}$ or $\{X_1, X_6, X_7\}$ or X_1 and X_6 should be in a new factor.
- Attributes X_1 and X_6 are complementary attributes, i.e. the DM prefers an outcome where both attributes are present.

Consequently, if the preference ordering when $p = 0.5$ were as follows $L_2 \succ L_1$, we can increase the probability p by half to 0.75 and ask the DM the same comparison gamble query. This time, the DM prefers an outcome where attribute X_1 may act as a substitute for attribute X_6, and vice versa if her preference is now $L_1 \sim L_2$.

Consider the following scenario: a DM is queried a comparison gamble question (regarding the GAI network in Figure 3 (page 13)) about attribute X_1 and X_2; she yields a probability of 0.3. However, when a comparison gamble question about attribute X_1 and X_3 is posed, the DM yields 0.9. Albeit all three attributes belong to the same factor, the theory does not make any predictions about the degree of dependence between the attributes. The factor can also be separated into two distinct factors where X_1 and X_2 belong in one and X_1 and X_3 belong in another.

We built our system to test our two main hypotheses:

Hypothesis 1: **On preference Elicitation:** It is possible to elicit the structure of a sound GAI network by having a minimally trained DM 1) create a value tree by following Value Focused Thinking and 2) map the domain attributes from an ontology into the fundamental objectives of the value tree (that are fulfilled) by using the interface developed.

Hypothesis 2: **On verification of the output of the elicitation procedure:** It is possible to measure the quality of the obtained GAI network by asking a (minimally trained) DM to answer comparison gamble questions comparing two lotteries.

To test Hypothesis 1, we would need to measure the efficiency of the user interface in aiding the DM to build her value tree using value-focused thinking concepts; and the ability of the user interface in supporting the DM to specify attributes from a domain ontology under objectives built in the value tree.

To test Hypothesis 2, we need to measure the quality of the GAI network structure by measuring the dependence between attributes in factors at differing number of edges. We employ the three properties of a GAI network assumed to comprise of additively independent factors.

- If two factors are disconnected, the probability between any two attributes in those factors should be 0.5 (i.e. dependency = 0).
- If two attributes belong in the same factor, the probability should not be 0.5 (dependent).
- Moreover, if two factors are connected, the probability between any two attributes in those factors should not be 0.5.

3.2. Non-Expert Decision Makers

The motivation for our approach was to aid non-expert decision makers to express their preferences with a naïve idea about fundamental and means objectives, attributes and entities within a domain ontology, and the fulfillment of objectives by means of attributes.

We designed and implemented a test-bed system keeping in mind that simplicity in the user interface with very little bloat would aid our users in using the system effectively. Because the concepts of value-focused thinking and ontologies are not every-day notions, a highly complicated system with many features would only strive to make a decision maker confused and inefficiently perform the tasks needed to express her value tree.

3.3. Real-Estate domain

We chose the real world problem of searching for an apartment to test our hypotheses. All the examples in the following chapters regarding attributes, objectives and ontologies will concern apartments. Real-estate is a relatively complicated test domain. Yet the domain is quite familiar to most DMs for which it makes sense to elicit a complex preference model.

Modeling the domain ontology for apartments was aided by using a formal ontology such as MILO (Mid-level ontology) [http://www.cim3.net/. 2007]. Having an upper ontology at our disposal, especially when it consists of a formal upper layer (SUMO) enriched with concrete domain-independent 'middle layer' concepts (MILO), aided the creation of a domain specific ontology for apartments. It allowed us to focus on the content of the domain specific ontology without having to worry on the exact higher structure or abstract philosophical framework that gives ontologies rigid backbones.

3.3.1. Apartment Ontology

Publicly available apartment ontology publication sites, such as SWOOGLE, SchemaWeb and OntoSelect etc were not suitable for our purposes because of the following issues:

Some have too "shallow" ontologies. For example, **APARTMENT-UNIT** entity and **APARTMENT-BUILDING** entity would be defined; however, the ontology did not delve much further into the domain in terms of subclasses or properties of these entities.

Others have only partial ontologies, i.e. several entities would be defined in a hierarchy but relations between these entities (interactions between one entity and another) were absent.

Fortunately, many aspects of the domain ontology we needed were found in the Mid-Level Ontology (MILO) as considerable research has been done for this ontology. MILO is meant as a "middle" ontology between an Upper ontology, in this case the Suggested Upper Merged Ontology or commonly known as SUMO [Niles, et al. 2001], and domain ontology such as an apartment's ontology. Therefore, the best approach was to create a domain ontology of apartments based on MILO and SUMO jointly. SUMO is a candidate for the IEEE standard for a formal upper ontology and was also a good starting point for the domain ontology.

An initial concept map was built based on significant elements of the domain that are necessary for expressing preferences over individuals within the domain of apartments. These elements were identified from popular apartment rental websites such as (http://www.rentsline.com/, http://www.rentbc.com/, http://apartmentguide.ca/, www.move.com etc). The domain ontology was built upon the concepts already defined in SUMO and MILO by importing these two ontologies.

Significant entities of the domain such as **FLOOR-TYPE** that were missing in both MILO and SUMO were added into the ontology. Such entities including any sub-classes and relations between other entities that were identified as relevant within the domain, were introduced into the domain ontology.

Individuals and relations between entities from the concept map that were absent were also added. For example, the *hasAppliance* property was not present in either MILO or SUMO, it would be an owl:DatatypeProperty of type Boolean for the entity **APARTMENTUNIT**.

There were certain entities and relations already created in MILO or SUMO; however, the relation between these entities and relations were absent. These were added in the domain ontology, for example the *hasApartmentUnit* property between an **APARTMENT-BUILDING** and an **APARTMENT-UNIT** was created to describe that an apartment building HAD apartment units.

Some entities such as **FACILITIES**, **APPLIANCES** etc which would aggregate all factilities (Parking-lot, Swimming Pool etc) and all appliances (bathtub, fridge etc) already defined in SUMO or MILO had to be created as subclasses of other entities.

Certain entities such as **ROOM** were partially filled. Some subclasses of such as entities (LivingRoom, Office etc) were not defined and were also added.

The apartment ontology that was created was a fuller and much richer version of the intersection between the entities relevant to apartments in MILO and SUMO. Extensive additions and modifications had to be made in order for all the relationships and entities to be fully embodied in the domain ontology.

The complete apartment ontology OWL description is available in Appendix N.

Chapter 4. USER INTERFACE TEST BED

4.1. OVERVIEW

This chapter overviews the user-interface test-bed that was used to test the two hypotheses. It also describes each part of the workflow diagram shown in Figure 6 in detail.

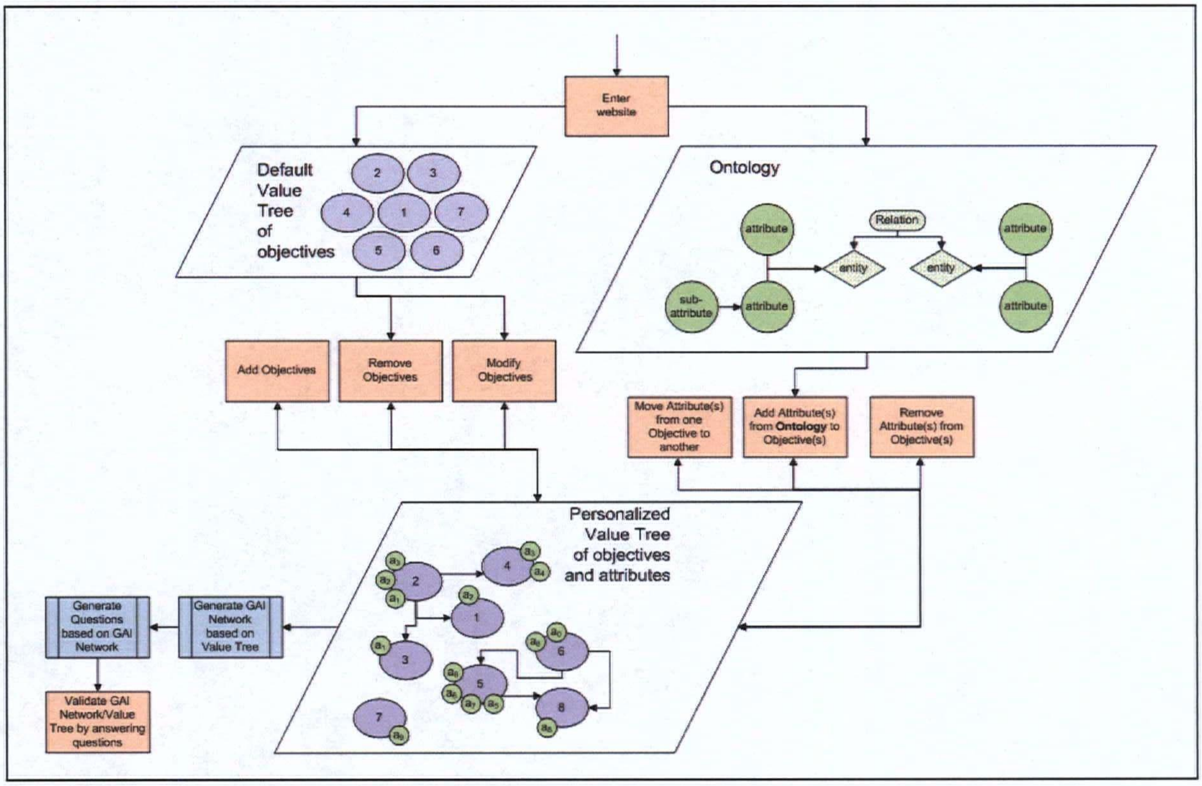


Figure 6: Workflow Diagram

The design of the user interface for the test-bed is primarily composed of four parts. Part 1 entails the visualization of the domain ontology. Part 2 entails the visualization of the default value tree. Part 3 entails the modification of the default value tree – this can be done by a) adding objectives, b) modifying the current objectives, c) removing objectives, d) adding attributes from the ontology in part 1 that fulfill the objectives in the personalized value tree and e) modifying the state of the attributes, i.e. the presence (true) or absence (false) of the attribute fulfills the objective. Once the DM has created a personalized value tree, we convert the tree into a GAI network as explained in Chapter 3. Part 4 comprises of testing the validity of the GAI network structure obtained by asking comparison gamble queries over the attributes used to describe the DM's value tree.

The user interface was written for the web-platform using Ruby on Rails [Hansson. 2004]. The chosen framework is quite adept at handling complicated user-interactions and is AJAX [Garrett. 2005] friendly. Therefore, every time a DM chose to perform an action on the value tree, a call to the server was seamless. Ruby on Rails' inherent Model-View-Controller design pattern allowed for the separation of the database and the interactions as well as the manipulations made to the data. This was necessary to keep the code clean and readable and the application versatile when it came to making changes.

4.2. Ontology Visualization

The ActiveRDF library was used for accessing the OWL data. ActiveRDF can be used as a data layer in Ruby on Rails providing a very convenient Object-Relational Mapping (ORM) to relational databases. Design and implementation of this application was sped up considerably by the simplicity of design of this library.

The domain ontology for apartments, in OWL format, was parsed into ntriples using the rapper utility. This was necessary since the ActiveRDF library only accepts ntriple formatted files for processing.

Using ActiveRDF library, the ontology was processed into a tree format of entities, the relations between entities and the domain and range values of these relations, and the attributes of each entity. For the purposes of simplicity only *:subPropertyOf* and *:subClassOf* relations were processed for visualization.

Using a JavaScript Tree Control library the ontology was visualized as shown in Figure 7. The DM is able to interact with all the leaf-level attributes only. The text of the non-allowed attributes and entities are italicized and grayed-out, thus enabling DMs to realize easily which attributes are interactive. These elements of the ontology are also static and non-draggable.

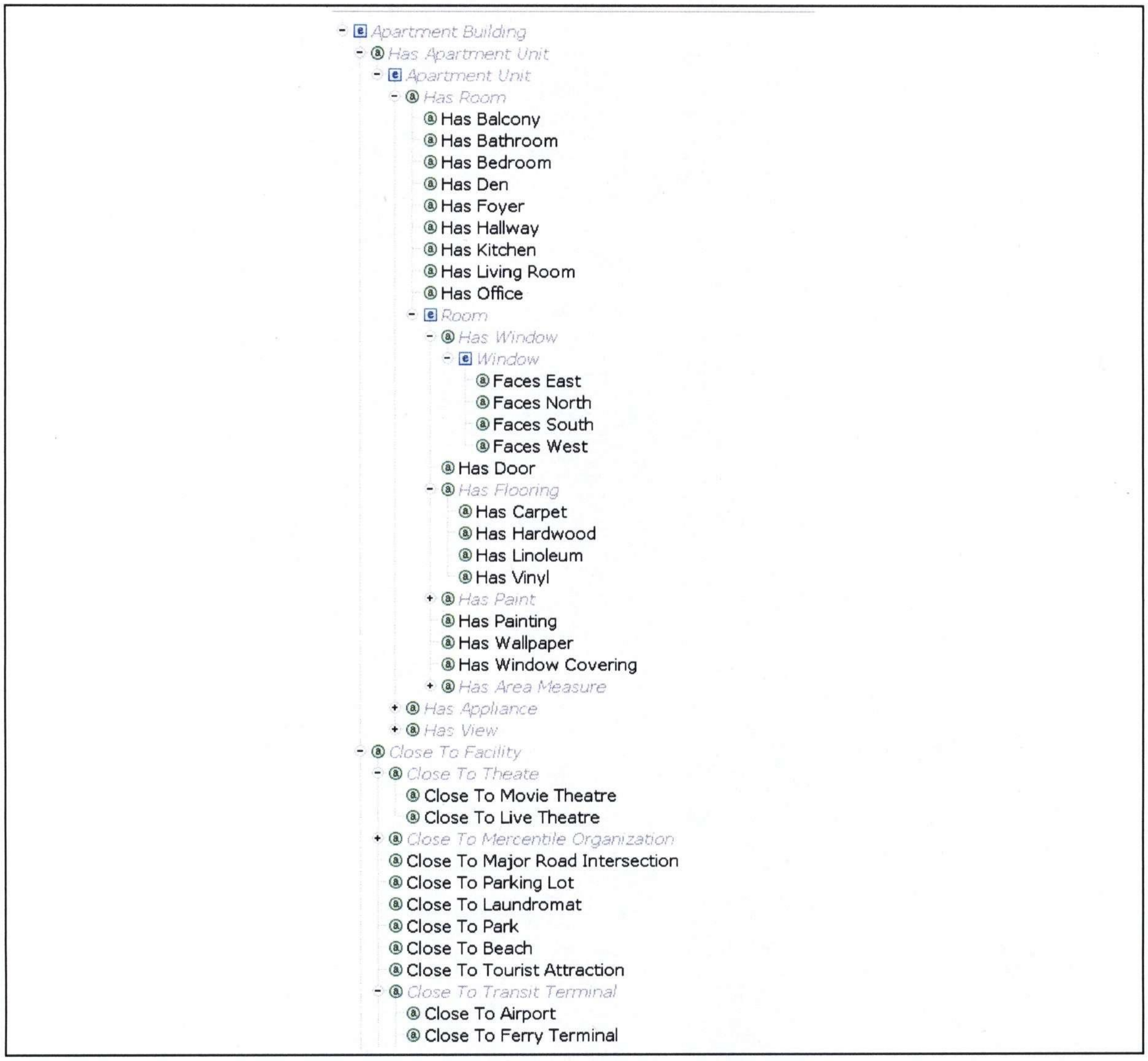


Figure 7: Sample Ontology Visualization

4.3. Value Tree Construction

The user interface provides each DM with a default value tree that specifies a set of objectives as shown in Figure 8. The default value tree is intended to facilitate the DM in expressing her fundamental objectives in the domain. The DM can look at the objectives and decide that the list is all encompassing of her own preferences and move on to specifying attributes that fulfill these objectives.

Figure 8: Possible default Value Tree in the Real Estate domain

A DM may also decide that the objectives in the default value tree only partially represents her preferences and can modify the ones that need to be changed. She may realize, while looking at the given value tree, that one or more of her objectives are missing and thus concentrate on identifying those. We argue that the default value tree reduces the mental effort of identifying a DM's objectives yet the interface allows her to modify and personalize the value tree to reflect her preferences if need be. However, a DM could also remove all the objectives and start from a blank value tree.

The user interface in addition to enabling a DM to identify her objectives, allows the DM to specify attributes that fulfill these objectives. When any action to modify the value tree is taken, a message describing that action is provided to the DM by means of a pop-up window on the screen.

4.3.1. Identifying objectives

A DM may add objectives to the value tree by specifying the statement in the text box shown in Figure 9. For example, if one of the user's objectives was to **MAXIMIZE STORAGE SPACE,** then to add this objective the DM would type this statement in the text box and click the Add button shown in Figure 9.

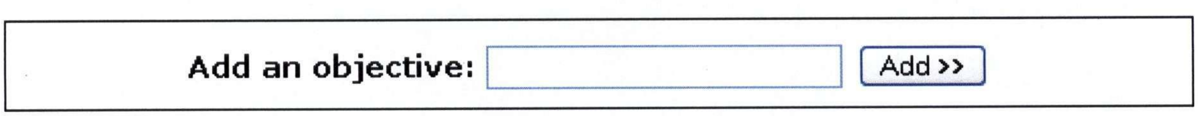


Figure 9: GUI Section for adding an objective to the value tree

Once added, the value tree will look as shown in Figure 10. To modify an objective in the value tree, a DM must click on the pencil icon [ ] directly left of the objective label. For example, if the DM wanted to modify the objective **MAXIMIZE THE COMFORT OF THE OUTDOOR LIVING SPACE** then he must click on the pencil icon [ ] next to this objective. When clicked, the objective label becomes an in-line editing text box with two options "OK" and "cancel". The DM may now input the new objective label, such as **MAXIMIZE THE COMFORT OF LIVING SPACE** in the in-line text box**.** Once the DM has completed modifying the objective label, he may click on "OK" to save the changes or "cancel" to reset the changes.

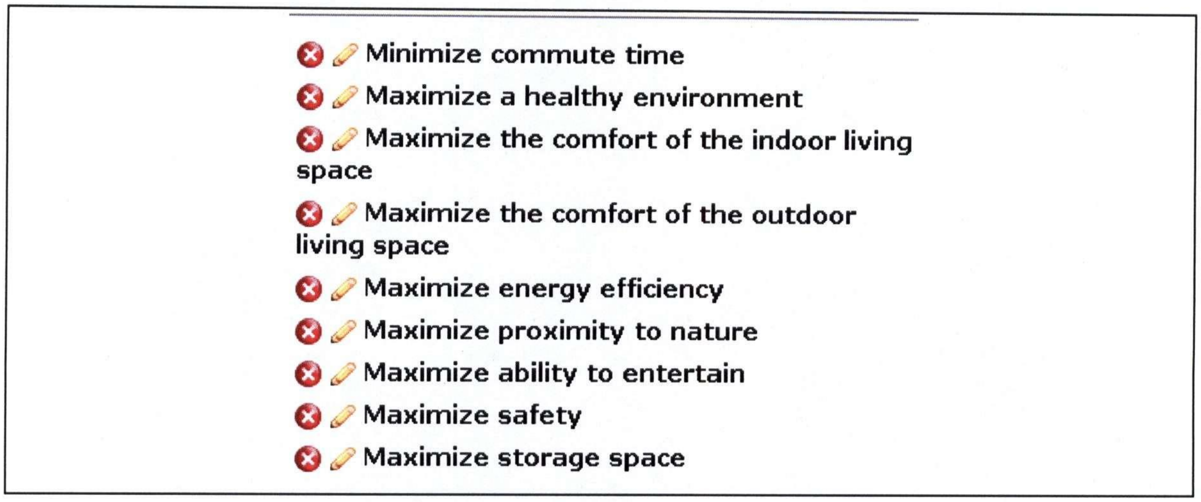


Figure 10: Personalized Value Tree

To remove an objective from the value tree the DM must click on the red-cross [ ] icon, which is to the left of the objective label. For example, if the DM wishes to remove the objective **MAXIMIZE THE COMFORT OF THE INDOOR LIVING SPACE,** he must click on the red-cross [ ] icon left of this objective. This will pop up a message asking the DM to confirm the removal of the objective from the value tree.

4.3.2. Mapping attributes into Objectives

To specify attributes that fulfill objectives, a DM must select the attribute from the ontology on the right-hand side and drag the attribute under the objective. For example, the objective **MINIMIZE COMMUTE TIME** may be fulfilled by the attribute *Close To Bus Terminal.* The DM must select the attribute *Close To Bus Terminal* and drag the attribute on the highlighted box for **MINIMIZE COMMUTE TIME**, as shown in Figure 11.

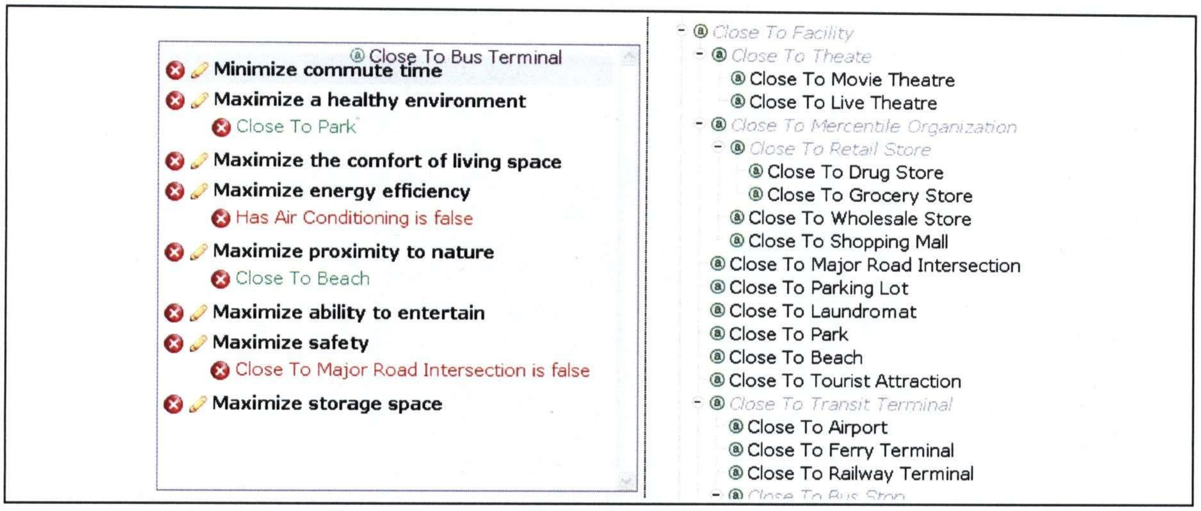


Figure 11: Adding an attribute to an objective

If an attribute is not dragged under any objective, then the attribute is not added to any objective in the value tree. Once the attribute is added to the objective, the attribute shows up in green under the objective label as shown in Figure 12. The color green indicates that the value of the attribute should be true to fulfill the objective. For example, to fulfill the objective minimize commute time the DM would like to have an apartment close to the bus terminal. However, if a DM wants to specify that an attribute satisfies an objective when its value is false, she has to add the attribute to the objective and then double click the attribute. For instance, to maximize energy efficiency, the DM would like the apartment to have no air conditioning. The DM must add the attribute *Has Air Conditioning* under the objective **MAXIMIZE ENERGY EFFICIENCY** then double clicks on the attribute. The attribute will change to *Has Air Conditioning is false* in red, specifying that when the apartment does not have air conditioning, the objective **MAXIMIZE ENERGY EFFICIENCY** is fulfilled. Note, the DM can add the same attributes to several objectives. For instance, the attribute *Close to Park* may fulfill the objectives **MAXIMIZE A HEALTHY ENVIRONMENT**, as well as **MAXIMIZE PROXIMITY TO NATURE** as shown in Figure 12.

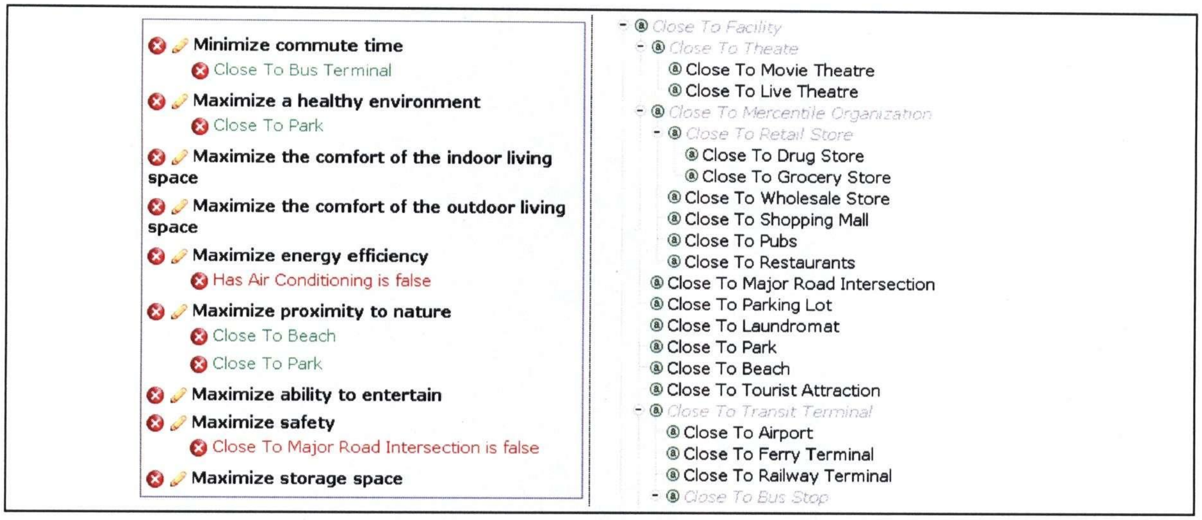


Figure 12: After adding an attribute to an objective

A DM may also move an attribute from one objective to another by dragging the attribute label under the source objective to under the label of the destination objective. For example, the value tree shown in Figure 12 illustrates that the objective **MAXIMIZE A HEALTHY ENVIRONMENT** has the attribute *Close To Park*. If the DM decides that this attribute is more suitable to fulfill the objective **MAXIMIZE PROXIMITY TO NATURE**, then the DM can drag the attribute *Close To Park* from the objective **MAXIMIZE A HEALTHY ENVIRONMENT** to **MAXIMIZE PROXIMITY TO NATURE**.

A DM may also decide at a later point that an attribute added under an objective, does not really fulfill the corresponding objective and can remove it by clicking on the [⊗] red cross icon to the left of the attribute label under that objective. For example, if a DM were to decide that being close to a beach did not really fulfill the objective of maximizing the proximity to nature due to the large crowds of people that gather during summer time, the DM would click on the red cross icon on the left of the attribute label *Close To A Beach* which would pop-up a message asking to confirm whether the DM really wishes to remove the attribute *Close To A Beach* from the objective **MAXIMIZE PROXIMITY TO NATURE**.

One useful feature in the user interface is a means to easily locate the position of all attributes under an objective in the ontology by clicking on the objective label thus highlighting the objective and all the attributes under the objective in the Ontology to its right. For example, if the DM had a value tree as shown in Figure 13, the DM would click on the objective label **MAXIMIZE A HEALTHY ENVIRONMENT** to highlight the attributes *Has Gymnasium, Has Swimming Pool, Close To Swimming Pool, Close To Gymnasium, Close To Baseball Field,* *Close To Park, Close To Beach* in the ontology.

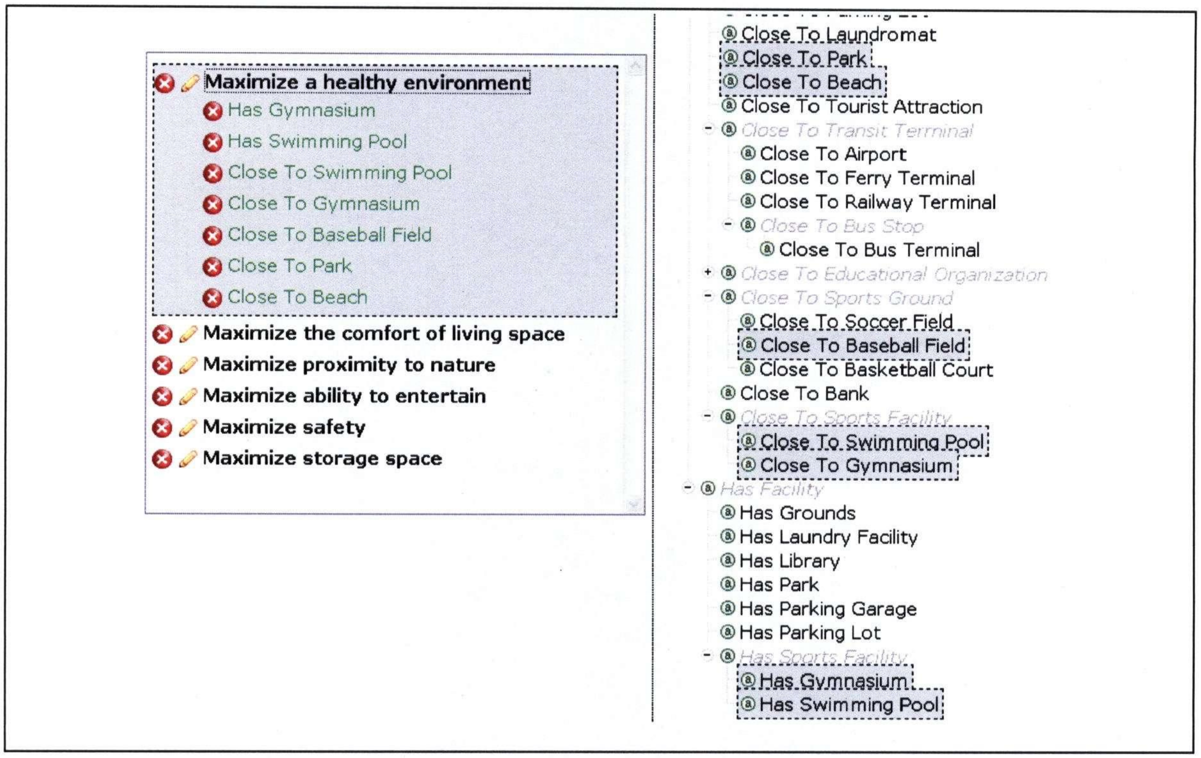


Figure 13: Attributes under objective highlighted in Ontology

4.4. VALUE TREE TO GAI NETWORK

Once the DM is satisfied with her value tree, she would click on the search [ ] icon. The value tree is converted into a GAI Network, where the objectives are factors and the attributes to fulfill each of the objectives are the attributes for each factor*.

The GAI network is created using a graphing library from Ruby called GRATR. For example, the value tree as illustrated in Figure 14 is converted into the GAI network shown in Figure 15 (this GAI network is not available for the DM but is visualized for the purposes of this thesis).

* For more information, see Chapter 3.

I_{174}: Minimize commute time
- a_{63}: Close To Grocery Store
- a_{78}: Close To Railway Terminal
- a_{79}: Close To Bus Terminal
- a_{89}: Close To College

I_{175}: Maximize a healthy environment
- a_{73}: Close To Park
- a_{103}: Close To Baseball Field
- a_{107}: Close To Swimming Pool
- a_{108}: Close To Gymnasium
- a_{5}: Has Balcony

I_{176}: Maximize the comfort of the indoor living space
- a_{40}: Has Dishwasher
- a_{17}: Faces South
- a_{22}: Has Hardwood
- a_{11}: Has Kitchen
- a_{5}: Has Balcony

I_{177}: Maximize the comfort of the outdoor living space
- a_{114}: Has Park
- a_{111}: Has Grounds

I_{178}: Maximize energy efficiency
- a_{37}: Has Air Conditioning is false
- a_{17}: Faces South

I_{179}: Maximize proximity to nature
- a_{74}: Close To Beach
- a_{73}: Close To Park

I_{180}: Maximize ability to entertain
- a_{118}: Has Swimming Pool
- a_{5}: Has Balcony

I_{181}: Maximize safety
- a_{70}: Close To Major Road Intersection is false
- a_{47}: Has Security Alarm

Figure 14: Sample Personalized Value Tree

The library takes in an array of attributes for each factor and creates the edges based on the common attributes. For example, for the objectives **MAXIMIZE THE COMFORT OF INDOOR LIVING SPACE** and **MAXIMIZE ENERGY EFFICIENCY** from the Value Tree illustrated in Figure 14, the common attribute *Faces South* causes an edge to be created between the nodes Factor 178 and Factor 176 in Figure 15. Factors that have no common attributes show up as independent nodes such as Factor 174, Factor 177 and Factor 181.

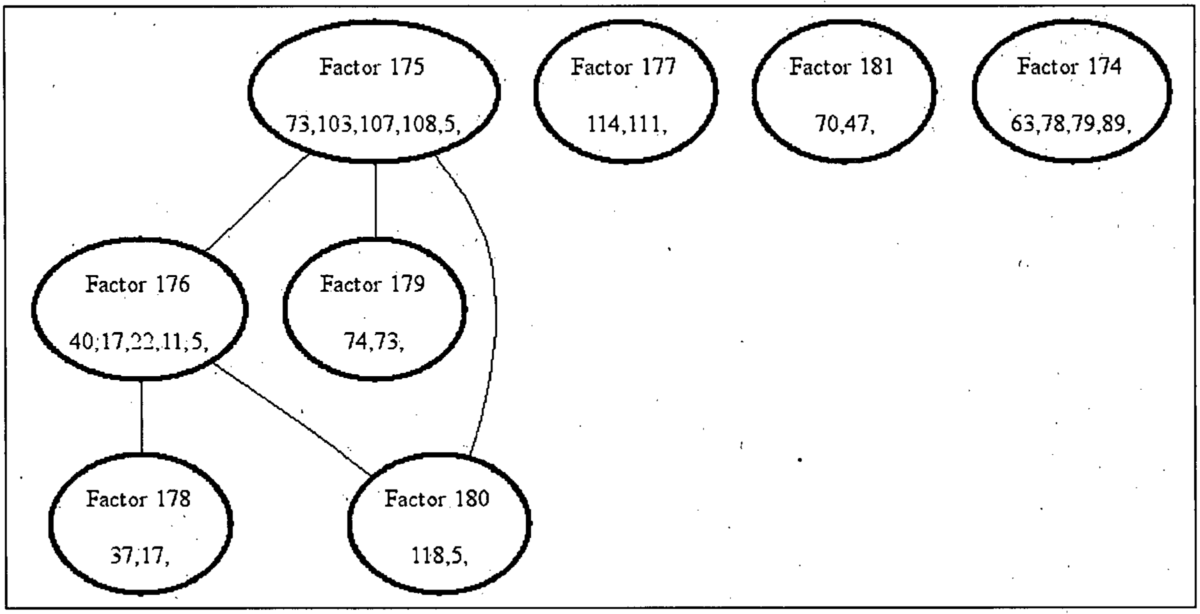


Figure 15: Sample Personalized GAI network

4.5. Measuring the quality of the GAI Network

Once a DM expresses her value tree, and a GAI network is generated, a set of questions are created to validate how well the GAI network represents the DM's true preferences. We are interested in measuring the degree of independence between attributes within objectives in the value tree. To measure this independence value we propose to the DM a comparison gamble question as generated from the algorithm below.

4.5.1. Comparison Gamble

The comparison gamble consists of paired comparisons in which the DM must choose between two alternatives (See Section 2.3.2).

```
for distance = DMIN to DMAX
  Randomly (avoiding possible reuse of attributes) pick two
  attributes AK,AJ such as distance(factor(AK), factor(AJ)) =
  distance
  Propose the comparison gamble
    Option A) 50*  (AK-best, AJ-best) : 50(AK-worst, AJ-
  worst)
    Option B) 50  (AK-best, AJ-worst): 50(AK-worst, AJ-best)
```

We assume that D_{MAX} is the maximum possible distance between two attributes in the GAI Network, so for the GAI graph in Figure 15, D_{MAX} is 3. We assume that D_{MIN} is the minimum possible distance between two attributes in the GAI Network. In cases where the graph is fully connected and has no independent factors, D_{MIN} is 0: the shortest distance between two attributes is when they belong in the same factor. Whereas in a GAI graph, as shown in Figure 15 D_{MIN} is -1, due to the presence of non-connected factors.

Given a pair of attributes (A_K,A_J) in factors I_1 and I_2, the distance between the attributes in A_K and A_J is measured as the smallest number of the edges it takes to traverse from factor I_1 to factor I_2. For example, distance between attribute 37 and attribute 73 in Figure 15 is 2, not 3 even though there exists a path from two factors which have attributes 37 and 73 where the distance would be 3. Independent factors (factors that are not connected to any other factors) are a special case; with distance from all other factors being equal to -1†. For example, distance from attributes 70 and 47 in Factor 174 to any other attributes are -1.

The algorithm iterates from D_{MIN} to D_{MAX}, and searches through the graph for a pair of attributes where the shortest distances between them equal *distance*, then proposes the comparison gamble question as a paired comparison between two alternatives. In our case, the two alternatives are a lottery between two possible outcomes. One alternative, option A, initially provides the DM with the following outcomes: a 50% chance of being in a situation where the DM will receive the best of both attributes and a 50% chance of being in a situation where the DM will receive the worst of both attributes. Consequently in the next alternative, option B, the same outcomes are always provided for the relevant question. A 50% chance of being in a situation where the DM will receive the best of attribute A_J and the worst of

* This probability value is referred to as p in the comparison gambles described below.
† -1 was used to denote no path. It was a design decision made for simplifying implementation.

attribute A_K and a 50% chance of being in a situation where the DM will receive the best of attribute A_K and the worst of attribute A_J.

If the DM states she is indifferent between options A and B when the *p* probability is 50, the attributes in the pair are considered independent. However, if the DM is not indifferent we keep asking questions to achieve a narrower dependency estimate by adjusting the probability *p* in the comparison gamble. For instance, if the DM prefers B (the attributes are substitutes), we make A more desirable by increasing the probability at *p* to the midpoint of [50-100] which is 75. Now the DM may prefer A, therefore to get closer to identifying the indifference the probability at *p* is decreased to the midpoint of [75-100] which is 63. If the DM is not indifferent, depending on the choice, we will know that the indifference is between either 50 and 63 or 63 and 75. The DM is asked the same question with a different probability at *p* until she claims to be indifferent or the difference between the maximum and the minimum probabilities in the range is 1.

We claim that our estimate for indifference between attributes in a pair corresponds to an estimate for dependency between the attributes in the pair (the further the probability *p* is from 0.5 the more the two attributes are dependent)*. We validate the GAI network based on the observed dependence values of the attributes in the pair of each comparison gamble question proposed.

4.5.2. Validating The GAI Network

A comparison gamble question asked in the user interface's validation part is shown in Figure 16. Option A is a choice with two outcomes: either a good outcome (for e.g., presence of dishwasher and facing south) or a bad outcome (i.e. absence of dishwasher and not facing south).

* See Section 3.1 for more details.

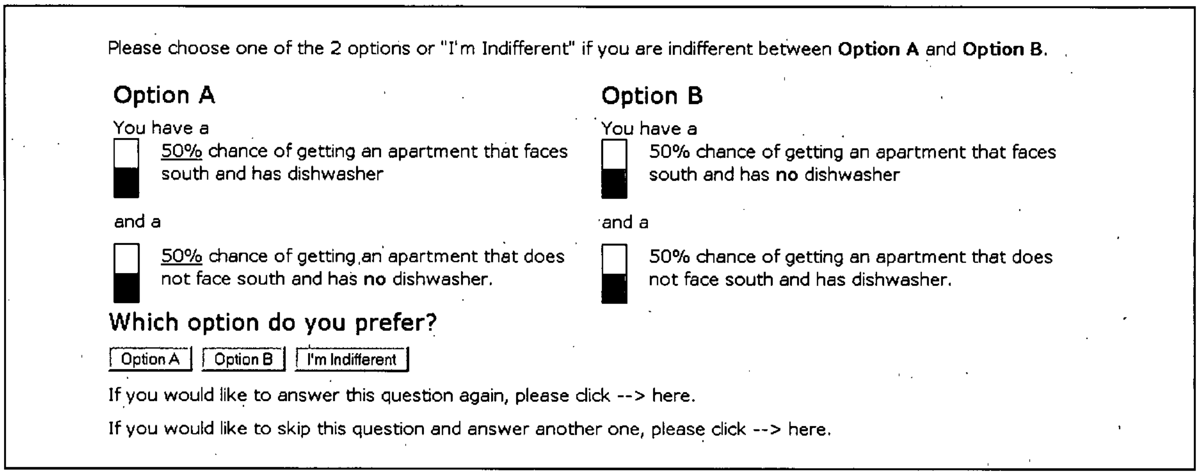


Figure 16: Sample Comparison gamble Question

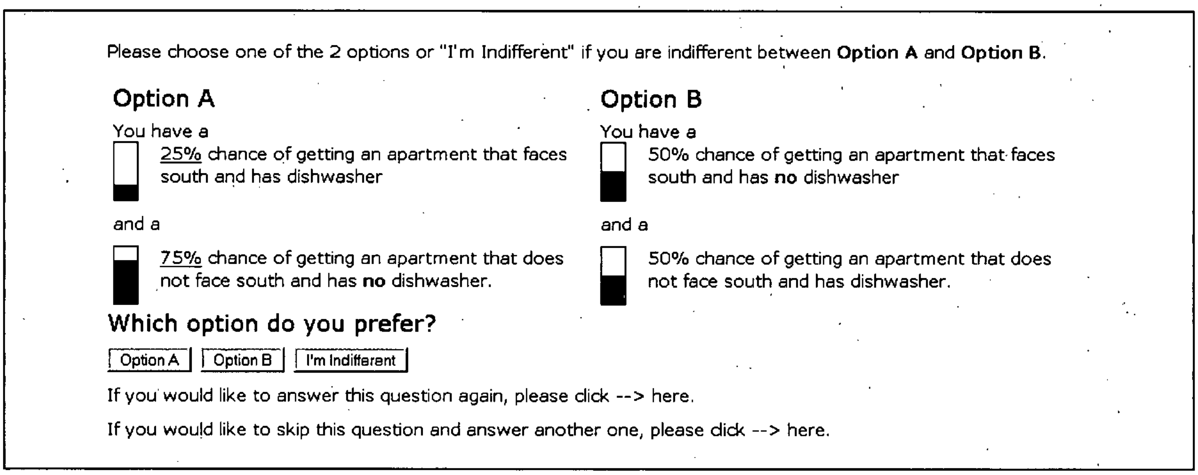


Figure 17: First time Option A chosen for Comparison gamble Question

Option B is an intermediate in desirability between the good and bad outcomes of Option A, i.e. presence of dishwasher and absence of a south facing apartment and vice-versa. DMs are asked to choose between the gambles Option A and Option B or whether they are indifferent given the current set of circumstances. If the DM chooses Option A, the probability of the good outcome in Option A decreases to make Option B more worthwhile as shown in Figure 17, and vice versa. This probability is varied until the DM is indifferent between the two alternatives. The DM can express indifference in two ways, by choosing the "I'm Indifferent" button or by choosing the options until the difference between the two probabilities in Option A is 1. If a DM makes a mistake, she can click the link to answer the question again. The DM may also skip a question for whatever reason if they are uncomfortable with the question or choose not to answer it by clicking on the link for skipping that question.

Chapter 5. EXPERIMENT

5.1. GOAL

We performed a series of experimental studies to test our approach for eliciting a DM's GAI network. We will first discuss the set of hypotheses which we are testing. Then we will describe in detail the experimental study we ran to test our approach; this will include a discussion of the methodology employed, the participants involved, the procedures carried out, the data collected and analyzed. In conclusion, the results are discussed and some future work is proposed.

5.2. HYPOTHESES

We have two major hypotheses and each of these hypotheses requires several sub-hypotheses. They are detailed below:

Hypothesis 1 On preference Elicitation: It is possible to elicit the structure of a sound GAI network by having a minimally trained DM 1) create a value tree by following Value Focused Thinking and 2) map the domain attributes from an ontology into the fundamental objectives of the value tree (that are fulfilled) by using the interface developed.

Hypothesis 1.1 A DM would understand the concepts of Value-Focused thinking from minimal training. (10 minutes*)

Hypothesis 1.2 A DM would understand the concepts of the hierarchical nature of the domain represented as an ontology from minimal training. (10 minutes*)

Hypothesis 1.3 The default value tree would help the DM in building her personalized value tree.

Hypothesis 1.4 A DM is capable of building her personalized value tree by identifying fundamental objectives autonomously.

Hypothesis 1.5 The GAI network obtained from the mapping is sound.

Hypothesis 1.5.1 Attributes in the same factor are additively dependent while attributes in disconnected factors are additively independent.

---

* See Appendix E.

Hypothesis 1.5.2 The degree of dependency between pairs of attributes decreases proportionally to the distance d between the pairs of the attributes, where given two attributes (a,b) in the pair, distance d between the attributes denotes the shortest number of edges spanned between the containing factors of a and b.

Hypothesis 2 On verification of the output of the elicitation procedure: It is possible to measure the quality of the obtained GAI network by asking a (minimally trained) DM to answer comparison gamble questions comparing two lotteries.

Hypothesis 2.1 A DM understands the comparison gamble questions comparing the two lotteries given minimal training (10 minutes).

Hypothesis 2.2 A DM can answer many comparison gamble questions in sequence without losing performance/accuracy/consistency in her answers.

5.3. METHODOLOGY

We have designed a user study to verify Hypothesis 1, the key hypothesis on which our preference elicitation approach relies. However, we needed to perform a pre-study to validate the quality of the default value tree, which was needed for the user-study. The methodologies for both studies are described below. We call the pre-study, S_0 and the actual user study S_1.

5.3.1. PRE-STUDY S_0

A study, S_0, with twenty-eight participants, was conducted to validate the quality of the default value tree. This was necessary so that participants for study S_1 could begin with a value-tree that was general enough that it would not require too-much thought when the participants begin. In S_0, participants were sent a survey through email to answer the following question:

> *... Suppose you were searching for an apartment. You want the place that you are searching for to fulfill certain goals and objectives you have, such as, you want to reduce the amount of time it takes for you to commute to either work/school from home - so you want to "minimize commute time".*
>
> *... If you could please take a look at the following list and let me know whether*
>
> *something should be removed (which number)*
>
> *something should be added (which number)*
>
> *the wording needs to be changed so that it is clear immediately what is being talked about (which number)*
>
> *... The goal is to find out whether I've been able to capture the majority of things people*

care about when searching for a new place ...
This is the list of fundamental objectives:

Minimize commute time

Maximize a healthy environment

Maximize the comfort of the indoor living space

Maximize the comfort of the outdoor living space

Maximize energy efficiency

Maximize proximity to nature

Maximize ability to entertain

Maximize safety

Based on their answers, the most popular remarks and the most general objectives were compiled to form a default value tree as shown in Figure 8*.

5.3.2. User Study S_1

In S_1, the goal of the study was to evaluate the effectiveness of our approach in eliciting a DM's preferences expressed as a GAI network. Two pilot studies were performed to validate the format of the user study and the final user study was conducted using ten participants. The participants for the pilots were not remunerated and were friends of the author. The participants for the study S_1 included graduate students, two females and all between the ages of 20 - 40. The procedure for the study S_1 is detailed below†.

5.4. Procedures

Each participant was asked to sign a consent form‡. After that, she was asked to fill out a brief questionnaire where demographic information such as age, gender etc were collected§.

5.4.1. Training

Participants needed training before the tasks in the user study could be performed. The training comprised of two parts. Training Part 1 began with participants being given a brief

* See Section 4.3.
† See Appendix G.
‡ See Appendix H.
§ See Appendix I.

introduction of the research question and relevant concepts such as Value Tree, Objectives, Attributes and Ontology, which were explained in the first person using a sample domain for office desks † (these exact terms and phrases were used).

> *When objectives are organized hierarchically, they form a value tree. On the left side is a value tree. I have been provided with a value tree to begin with. I can personalize this value tree to reflect my preferences.*
>
> *My objectives can be fulfilled by things in the real world, i.e. things such as 'made of wood', 'wide keyboard drawer' etc. We call these elements in the real world, attributes. Most often attributes belong to a class that is a set of objects, known as entities such as 'desk', 'drawer' etc. These entities and attributes often have a hierarchical representation in the real world – we call this domain ontology.*

Each participant was then asked to train on the desk domain for each of the following actions*:

- Adding an objective
- Modifying an objective
- Removing an objective
- Adding an attribute to an objective
- Changing the attribute value under an objective from true to false
- Moving an attribute from one objective to another
- Removing an attribute from an objective

Once the participants completed this set of actions, Training Part 1 concluded.

For Training Part 2†, each participant was given a brief introduction to the concept of a standard gamble question:

> *You will be asked to determine probabilities that make you indifferent between pairs of outcomes. Outcomes can be certain or uncertain. In addition, they can be gains or losses. ... Here is a sample question.*
>
> *Please determine the value of p* ***<u>that would make you indifferent</u>*** *between* ***OPTION A*** *and* ***OPTION B.***
>
> ***OPTION A:*** *to enter a lottery in which you have a p chance of winning* ***a free coffee and a muffin*** *and a (1-p) chance of winning* ***nothing.***

---

* For more details on each action, see Chapter 4.
† See Appendix J.

OPTION B: *to win a free coffee.*

Please write down all the values of p that you consider and your corresponding preferred outcome.

p	*1-p*	*I prefer ...*
0	1	**OPTION B**
1	0	**OPTION A**
0.5	0.5	

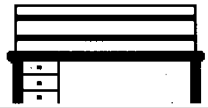
It was explained to DMs, when one preferred muffins and coffee, given a probability of 0 for p, i.e. winning nothing, the rational DM would choose Option B; although, a probability of winning both muffin and coffee, i.e. p is 1, causes a rational DM to choose Option A. However, when the probability of winning a free coffee and a muffin is 0.5, what option would the DM choose? When asked this question, participants proceeded to answer the simple standard gamble over a lottery and a sure outcome. When a participant would choose Option B, the p value would be increased to make Option A more favorable; however, when Option A was chosen the p value was decreased. The participants were given increments/decrements of 0.1 at each step of the process. When a participant would change their choice from the previous answer, the process would be concluded and the p value would be recorded as the arithmetic mean between the last p value and the second-to-last p value. This p value was the probability at which the participant was indifferent between the two options.

Following this step, participants were asked to answer a set of three standard gamble questions (comparing a lottery and a sure outcome), followed by a set of three comparison gamble questions (comparing two lotteries) in the office desk domain on paper.

The simple gamble questions had the following format:

p	1-p	I prefer ...
0.5	0.5	

For which value of p would you be indifferent between ***OPTION A*** *and* ***OPTION B****? Again, please write down in the table all the values of p that you consider and your corresponding preferred outcome.*

	OPTION A	OPTION B
p	Desk with 3 drawers and 3 shelves ☺ 	Desk with 1 drawer and 1 shelf ☺ 
1-p	Desk with no drawers or shelves ☹ 	

p that makes you indifferent ______________________

Participants were given a simple visualization of the standard gamble in the form of a table where the first column represented probability, the second and third column represented the two alternatives. The rows of the table usually represented a lottery in the alternative.

The participants were asked to answer these questions on their own, however assistance was provided when asked and answers were verbally confirmed to represent what the participant was trying to express. Therefore, for the above question, if a participant's indifference probability was at 0.75, it could be that the participant preferred some storage space rather than gamble and end up with none.

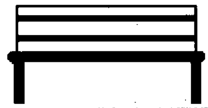
The comparison gamble questions followed the following format:

Let us focus on the desk domain. You are searching for a new desk. You are trying to come up with the best balance for your storage needs. Assume that a large desk is best suitable for your needs, a medium desk will be a tradeoff and a small desk is going to be difficult to work with. The grid on the left hand side now has an extra column for ***OPTION B****. The percentages for* ***OPTION B****, however, are always the same, 50%. Consider two aspects of a desk. One is the* ***number of shelves*** *and the other is the* ***size of the desk****.*

p	1-p	I prefer ...
0.5	0.5	

For which value of p would you be indifferent between ***OPTION A*** *and* ***OPTION B****?*

Again, please write down in the table all the values of p that you consider and your corresponding preferred outcome.

	OPTION A	OPTION B
p	Large desk with 3 shelves ☺ 	Small desk with 3 shelves 
1-p	Small desk with no shelves ☹ 	Large desk with no shelves 

p that makes you indifferent ________________________

Once again, the participants were asked to answer these questions on their own and answers were verbally confirmed to make sure that participants were comfortable with the format of the questions and understood what it meant when particular options were preferred.

Once the participant had completed Training Part 2, she was introduced to the comparison gamble feature of the user-interface. This procedure is detailed in Section 4.5.2.

5.4.2. Task

Having completed both training parts, each participant was provided with the following scenario:

> *Suppose, you are relocating to Vancouver, British Columbia Canada. You have preferences when it comes to renting an apartment. You are given a fixed budget of $1000 for rent. For the purposes of the study, you do not need to consider minimizing your costs as an objective.*

Participants were then asked to create their own value tree in the domain of apartments. After having completed this part, they were asked to proceed to the validation part of the interface. The validation of the participant's generated GAI network based on the Value tree created in Part 1 took a maximum of fifteen minutes. DMs were asked to answer comparison gamble questions starting with an attribute pair that were from factors that were not connected, to pairs of attributes within the same factor and then increased the distance between the pairs of attributes to the maximum possible, after which this cycle repeated until the maximum amount of time had been reached. DMs were allowed to skip any question they felt too uncomfortable to answer or did not understand its phrasing. Once they had completed both parts, the study was concluded. Participants were finally provided with a questionnaire to express any suggestions, comments or questions they may have*. The study was concluded by asking participants to sign a receipt for the payment of remuneration.

5.5. Data

Each participant's value tree, generated GAI network and scatter plots based on the answers obtained from the questions asked in Part 2 of the study are present in Appendix A. Appendix B. and Appendix C.

5.5.1. Collection

Each participant's scatter plot shows the relationship between the distance connecting a pair of attributes A_i and A_j in the GAI network and the measure of dependence between them.

The dependence between a pair of attributes A_i and A_j is determined by the probability p, where $0 \leq p \leq 1$, that makes the DM indifferent between the two lotteries in the comparison gamble questions as described in Section 4.5.1. If p was 0.5, it indicates that the pair of attributes is additively independent. If $0.5 \leq p \leq 1$, the pair of attributes are dependent as substitutes, and if $0 \leq p \leq 0.5$ the attributes are complements.

The dependence in the comparison gambles was normalized to a number between 0 and 1, which consequently removed the information of the dependence classification, i.e. whether they were substitutes or complements.

The observed dependencies between each pair of attributes were then normalized into the measure of dependence$M_D = |O_D - 0.50|/0.50$, where M_D is the measure of dependence and O_D is the observed dependence p. The analysis of this data is discussed in more detail in the following section.

5.5.2. Analysis

The measure of dependence between attribute pairs (obtained in the validation phase) and the distance between attribute pairs in the GAI network was plotted as shown in Figure 18. Figure 18 also shows the interpolation-line for all the observed points in the scatter plot. The size of the dots on the plot represents the number of observations made for that measure of dependence at that distance. Based on the analysis of the data, we had to remove one

* See Appendix K.

participant from our results and analysis as it was clear from the value-tree and the GAI network generated that the participant did not grasp the task. In total, ten participants' data has been analyzed and discussed below.

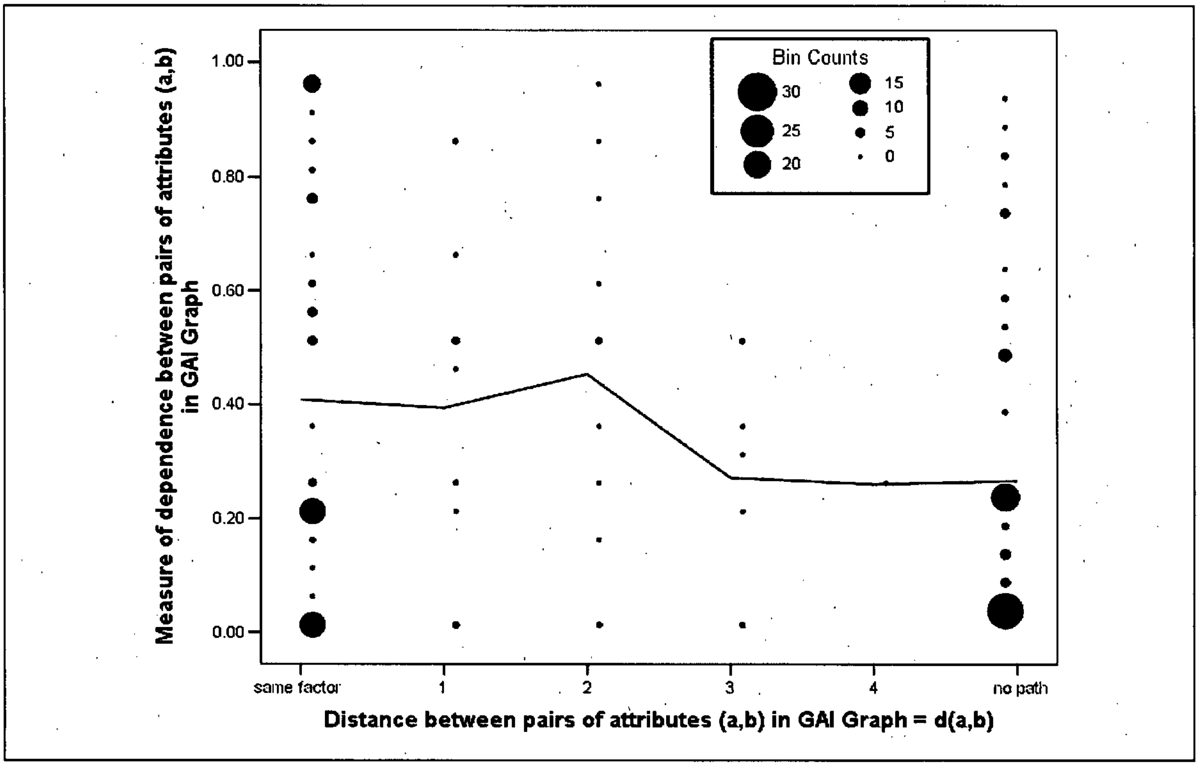


Figure 18: Scatter plot of all observed pairs of dependency between attribute pairs and distance between attribute pairs

Table 1 shows the same information presented in Figure 18, aggregating the ranges of the measure of dependence in a tabular format, which allows for precise reading. Please note there was only one participant (User 10*) who had answered one question where the distance between the pairs of attributes was 4.

---

* See User 11 at Appendix A. for further details

			Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)						Total
			same factor	1	2	3	4	no path	
Range of Measure of Dependence between pair of attributes (a,b) in GAI Graph	0.8 - 1.0	Number of Observations	15	2	2	0	0	5	24
		% within Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)	17.90%	14.30%	16.70%	0.00%	0.00%	5.60%	11.50%
	0.6 - 0.8	Number of Observations	12	1	2	0	0	7	22
		% within Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)	14.30%	7.10%	16.70%	0.00%	0.00%	7.80%	10.60%
	0.4 - 0.6	Number of Observations	10	5	3	2	0	13	33
		% within Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)	11.90%	35.70%	25.00%	28.60%	0.00%	14.40%	15.90%
	0.2 - 0.4	Number of Observations	24	3	2	3	1	23	56
		% within Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)	28.60%	21.40%	16.70%	42.90%	100.00%	25.60%	26.90%
	0.0 - 0.2	Number of Observations	23	3	3	2	0	42	73
		% within Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)	27.40%	21.40%	25.00%	28.60%	0.00%	46.70%	35.10%
Total Number of Observations			84	14	12	7	1	90	208

Table 1 : Total number of observations at each d(a,b) for a range of measure of dependence

After further analysis of the data, the observations were grouped into two categories, which represented the nature of the GAI network generated from the value tree: disconnected GAI graphs and connected GAI graphs. Five participants had disconnected graphs*, i.e. graphs with factors that have no edges between them as shown in Figure 19.

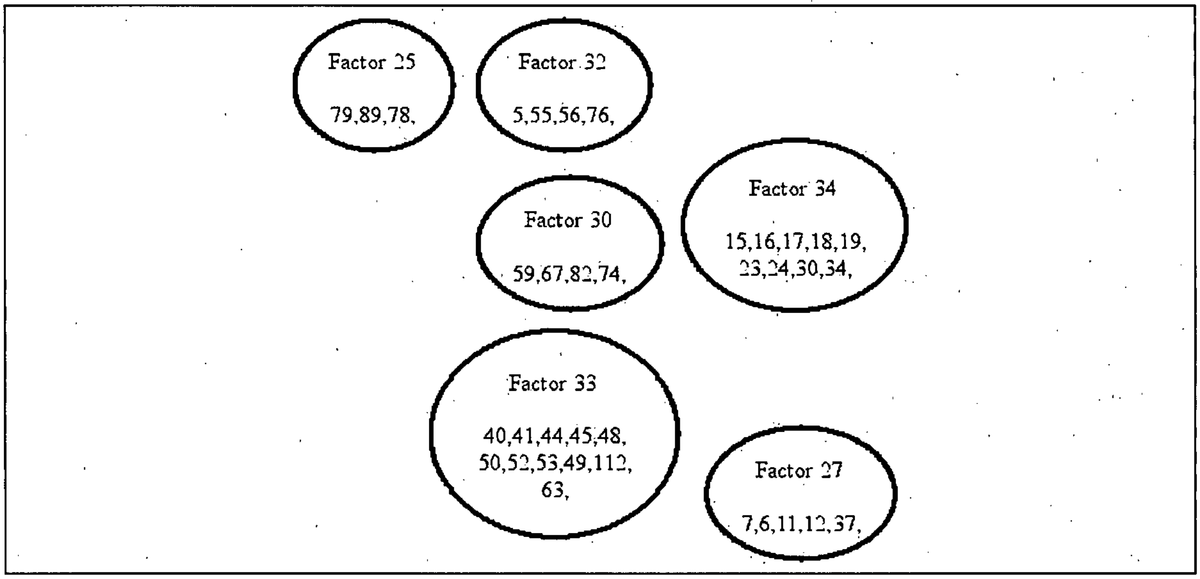


Figure 19 : GAI Network of User 1, exemplifying a disconnected GAI Graph

---

* User 1, User 3, User 5, User 6, User 7 and User 9 [See Appendix B. and Appendix C. for further details]

The other five participants had connected graphs*, where there were a minimum of seven factors and at least one edge between two factors were present, as shown in Figure 20.

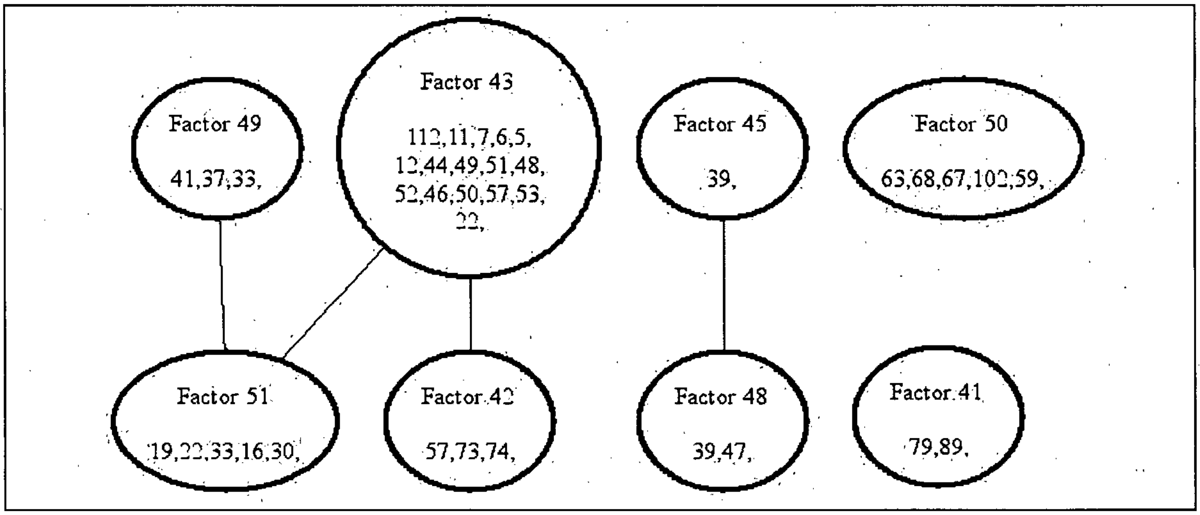


Figure 20 : GAI Network of User 2, exemplifying a connected GAI Graph

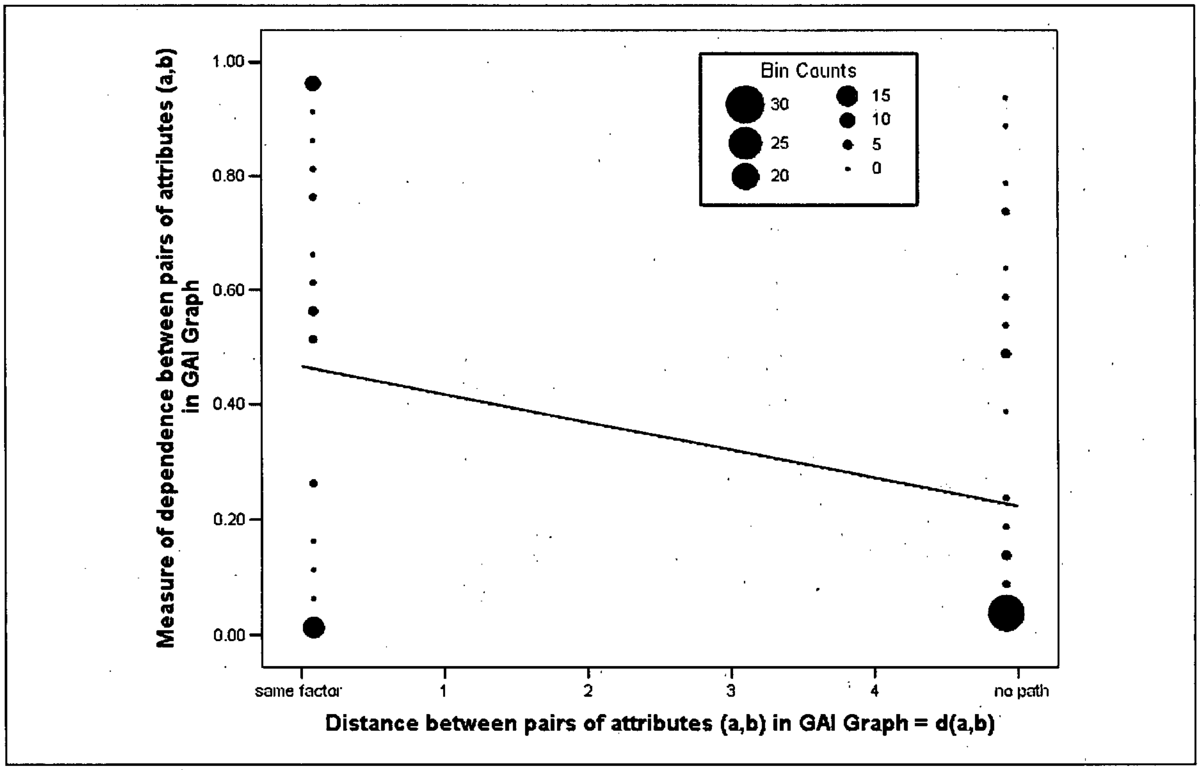


Figure 21 : Scatter Plot clustering of all observed pairs of dependency between attribute pairs and distance between attribute pairs on disconnected GAI networks

---

* User 2, User 4, User 8, User 10 and User 11 [See Appendix B. and Appendix C. for further details]

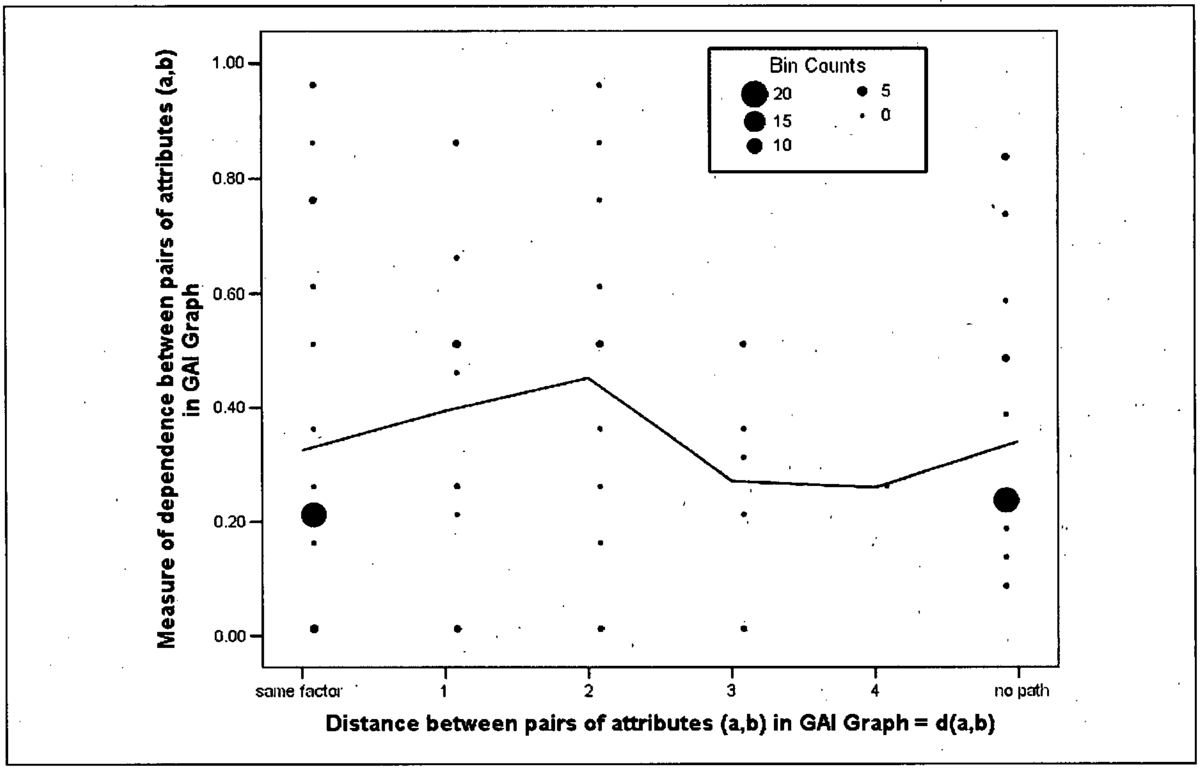


Figure 22: Scatter Plot clustering of all observed pairs of dependency between attribute pairs and distance between attribute pairs on connected GAI networks

Figure 21 shows a scatter plot of the measure of dependence between attribute pairs and the distance between attribute pairs for all disconnected GAI graphs. It also shows the interpolation-line for all the observed points in that case. Figure 22 shows a scatter plot of the measure of dependence between attribute pairs and the distance between attribute pairs for all connected GAI graphs also showing the interpolation-line for all the observed points in that case. Based on these figures, we notice that DM's with disconnected GAI nets followed the trend we predicted in Hypothesis 1.5.2; however, connected GAI nets did not follow any pattern.

The observations were grouped into another category, which represents the nature of the personalized value tree with respect to the default value tree. If the personalized value tree had all the objectives as the default value tree and no new ones were added or default ones removed or modified it was classified as an *unchanged default value tree*, otherwise the value tree was classified as *changed default value tree*. Table 2 shows the frequency of participants whose personalized value tree was either changed or unchanged. There was no significant relationship among the distance between pairs of attributes and the measure of dependence between them.

Nature of Personalized value tree	Frequency	Percentage
Changed Default Value Tree	7	70.00%
Unchanged Default Value Tree	3	30.00%
	10	100.00%

Table 2: Frequency of participants with state of personalized value tree

5.6. Discussion

Let us now discuss the results in terms of how they provide evidence for confirming or invalidating each hypothesis stated in Section 5.2.

Hypothesis 1 On preference Elicitation: It is possible to elicit the structure of a sound GAI network by having a minimally trained DM 1) create a value tree by following Value Focused Thinking and 2) map the domain attributes from an ontology into the fundamental objectives of the value tree (that are fulfilled) by using the interface developed". Hypothesis 1 is not verified by the above results: we discuss each of its sub-hypotheses for deeper insight into this claim.

Hypothesis 1.1 claims, "A DM would understand the concepts of Value-Focused thinking from minimal training. (10 minutes)". Every participant was given a very brief introduction to the concept of a Value Tree, Objectives and Attributes that fulfill objectives. It was also stressed to each participant that the default value tree provided did not necessarily reflect her objectives and was encouraged to add, remove and modify objectives from the value tree as she saw fit. Each participant was prompted to ask questions regarding any concept that they felt uncomfortable about during training. However, based on the design of the study and the results obtained it cannot be ascertained whether participants understood the concepts of Value Focused Thinking.

A compiled list of all the comments made regarding the value tree is shown below:

Quite clear and intuitive (User 2)

The tree covers pretty much what I need when I search for an apartment (User 9)

Well Selected Objectives for the Value Tree (User 5)

Good Objectives but a strong tendency to use the default value tree (User 1)

Perhaps provide verbal suggestions instead (User 1)

Perhaps a couple more default categories to get me thinking (too few makes me have to think about groupings) (User 6)

I did not use many of the default Value tree concepts. I wasn't thinking in terms of maximization as much as fulfilling different requirements of like the apartment will be used for parties, work, relaxation, cooking etc. (User 7)

Some general observations based on these comments are that some participants (2, 9, 5) were comfortable with the value tree and some (1, 6) were not. Table 2 shows that 70% participants' personalized value tree was different from the default value tree. This may indicate that most participants' personalized value tree was not well reflected in the default value tree; however, this may also indicate understanding of the concepts explained about Value Focused Thinking during training and thus personalization was easily done to reflect the participants' preferences.

Hypothesis 1.2 states, "A DM would understand the concepts of the hierarchical nature of the domain represented as an ontology from minimal training. (10 minutes*)" This hypothesis is confirmed. All participants showed a clear understanding of the concepts of domain ontology, entities and attributes introduced during training. There were no questions asked and no negative comments obtained during the study regarding the hierarchical structure of an ontology.

Comments from participants after the completion of the study regarding the ontology are as follows:

Clear and intuitive (User 2)

Needs more options and be broader (User 9)

Odd to have to add all the leaves when really, suppose, all I wanted was a window (not caring which way it faced) (User 1)

Do changes in the ontology affect a user's value tree? (User 2)

Maybe the ontology should have other types of attributes besides boolean (i.e. true and false) such as quantity or quality / thresholds (User 3)

Felt incomplete, such as no "bus stop" just "bus terminal" etc (User 5)

Some concepts such as doors and sinks are essential yet others are optional such as "close to park"; difficult to combine both. (User 7)

Again based on these comments some observations were made. Based on comment 1, it is clear that participant 1 was clearly able to grasp the concept of the hierarchical nature of the real world domain of apartments. Participant 1, however, wanted less entities and attributes to reduce the clutter and confusion present when first introduced to the ontology about apartments, whereas participants 3, 5 and 9 were less than thrilled with the limitations present in the ontology regarding the depth and the type of attributes present. The interface does not allow for any attributes that are not boolean, for example a quantity or a quality threshold etc such as number of bedrooms or quality of neighborhood. Participant 3 indicated that he was unable to express his preferences fully because of this missing feature. Absence of certain attributes in the ontology was another complaint that reflected itself in the usability of the interface for expressing preferences by participant 5. Participant 7 also mentioned a means to classify the significance of the attributes to a DM so that essential attributes such as *Has Sink* could be identified sooner over non-essential attributes such as *Close To Park*.

Participant 5 mentioned, while performing the tasks for the study that the concept of a hierarchy was strange for the value tree since the interface allowed one level for the objectives in the value tree. He was observed trying to create an objective beneath another and when asked what he was trying to do, the participant stated he would like to organize his objectives hierarchically.

Hypothesis 1.3 claims, "The default value tree would help the DM in building her personalized value tree." As seen from the above comments regarding the value tree, some participants (1, 2, 4, 5, 7, 8 and10) personalized their default value tree whereas participants 3, 6, and 9 were comfortable with the default value tree enough to express their preferences without having changed them. For instance, participant 1 may have grasped the concepts of objectives but was challenged in isolating his own value tree from the default value tree (as can be seen from the comment above) even though he did make modifications to his value tree. It is unclear whether DMs can identify their fundamental objectives without prompts from decision analysts and perform the task autonomously and thus Hypothesis 1.3 cannot be confirmed.

Hypothesis 1.4 states, "A DM is capable of building her personalized value tree by identifying fundamental objectives autonomously." Our study was not designed to provide clear evidence for either confirming or invalidating this hypothesis. W-e have obtained full value trees from each user but cannot conclude that these value-trees are structured so that they do not violate any independence assumptions.

Hypothesis 1.5 assumes "The GAI network obtained from the mapping is sound." To confirm or invalidate this hypothesis we need to delve a little bit deeper into its two sub-hypotheses.

Hypothesis 1.5.1 states, "Attributes in the same factor are additively dependent while attributes in disconnected factors are additively independent." This hypothesis is based on GAI network literature [Braziunas, et al. 2005, Bacchus, et al. 1995]. The cluster of observations under the same factor variable is hypothesized to be additively dependent, while those under the no path variable are hypothesized to be additively independent. A T-test was run on these two clusters and the significance obtained was 0.02. This hypothesis can thus be confirmed.

Hypothesis 1.5.2 states "The degree of dependency between pairs of attributes decreases proportionally to the distance d between the pairs of the attributes, where given two attributes (a,b) in the pair, distance d between the attributes denotes the shortest number of edges spanned between the containing factors of a and b.". This hypothesis is not verified, as is clearly illustrated in Figure 18. The function increases and then reaches a peak at distance 2 and then decreases sharply to distance 3 after which it levels out. This does not represent any pattern theorized in decision theory or graph theory literature.

Since not all sub-hypotheses for Hypothesis 1 were confirmed or invalidated, further study and research is needed for verification of Hypothesis 1.

Hypothesis 2 On verification of the output of the elicitation procedure: It is possible to measure the quality of the obtained GAI network by asking a (minimally trained) DM to answer comparison gamble questions comparing two lotteries." This hypothesis claims we can measure how sound a GAI network is based on the DM's answer about comparison gamble questions of attribute pairs. This hypothesis can be investigated in two-fold.

Hypothesis 2.1 claims, "A DM understands the comparison gamble questions comparing the two lotteries given minimal training (10 minutes)." Once participants received some training about the interface, they were given some constructive training about the structure of a standard gamble question comparing a lottery and a sure outcome and a comparison gamble question comparing two lotteries. A sample domain of desks was employed for this purpose. A cursory inspection of the answers was verbally confirmed from all participants when each section was completed. However, some participants (1, 7) stated they had difficulty in answering some of the questions while performing the study.

> *Structure of questions during validation was difficult. Would there be any other way of posing them? (User 1)*
>
> *Hard part is answering comparison gamble queries especially when one item is mandatory and one is superfluous such as 'sink' and 'near a park' (User 7)*

This can be taken in two ways. The participants understood the format of a comparison gamble question; however, questions that were proposed were challenging to answer. This is highly likely as can be seen from the comment above. Participants understood the consequence of answering such questions but were unaware of their true preferences in these regards. Alternatively, the participants did not understand the format of comparison gamble questions and were unable to answer them correctly. There is some evidence that this hypothesis is not valid.

Hypothesis 2.2 asserts, "A DM can answer many comparison gamble questions in sequence without losing performance/accuracy/consistency in her answers." The design of the experiment hindered us when trying to confirm or invalidate this hypothesis.

This analysis indicates that both hypotheses require further research. We may need to design specific experiments to test each hypothesis separately.

Chapter 6. CONCLUSIONS AND FUTURE WORK

This thesis explored a means of eliciting a DM's preferences in terms of a GAI decomposition. We delved into utility theory and the inherent structure of preferences; mentioned the independence assumptions made when preferences are structured; and described the background behind our decision in choosing the GAI decomposition to represent a DM's preferences. We further explained our motivation behind building a GAI network and the absence of methods for the elicitation of the GAI network structure. We proposed value-focused thinking as a viable approach to this problem and suggested using domain ontologies, which structure the vocabulary used to describe the world, in conjunction.

We experimentally evaluated a framework to support decision-making according to both a DM's objectives and a complex preference profile over a domain ontology in which the decision is contextualized. Our study reported that significant work in this area needs done and we suggest some viable options to consider that we were unable to do due to constraints in time and feasibility.

We have identified several aspects in which further research and study needs done to refine our approach and its evaluation.

The ontology presented to the DMs was trimmed out of all attributes that were not boolean. Further studies where attributes of all types are present may yield different results, as a DM will not feel limited in expressing her preferences.

We believe that using value focused thinking to elicit a DM's preferences should be explored further. DMs were only asked to consider their fundamental objectives when creating their value tree; however, if the value tree was more complete in terms of means and fundamental objectives, it is likely that the DMs would better understand the relevant concepts and be better able to express their preferences.

Perhaps, the attainment of a default value-tree can be done using a more thorough study. Some DMs suggested a longer value tree would be more useful to express their preferences, as they would find it easier to revise a large value tree than expanding a small one. However, one DM also suggested a list of fundamental objectives provided was distracting and caused him to think less about his own needs and goals when it came to searching for an apartment. It seems a possible solution that should be explored is to allow the DM to choose whether to start from an empty Value Tree, a small default one or a large default one.

We also believe providing the DMs with some guidance when eliciting a personalized value tree is necessary. This can be done by asking natural language questions based on the actual statement provided for a fundamental objective or while attributes are being added to the fundamental objectives. Perhaps after the validation part of the study, DMs could be given a rating of how well their answers compared to their value tree and give them an opportunity to re-arrange the value tree, or provide intelligent suggestions based on the answers provided as to what the predicted value tree looks like.

During elicitation of the generated GAI network, there could be an added step included where the objectives that make up the value tree and the attributes that fulfill these objectives are verified, giving DMs a chance to revise their value tree by providing an alternate means of visualizing the value tree. For example if a DM adds an objective **MINIMIZE NOISE POLLUTION** and places attributes such as *Close To Major Road Intersection is false*, *Has Garbage Disposal* and *Close to Grocery Store* under this objective, a fulfillment of this objective by means of these attributes could be assured. A possible solution may involve asking natural language questions where the DM is asked whether an attribute placed under an objective truly fulfills the DM's goal. Therefore, dependent attributes are left alone while attributes in dispute are identified as not withholding the DM's preferences. The DM is given intelligent suggestions to modify the value tree accordingly, possibly improving elicitation.

We believe supporting the integration of a DM's value with a domain ontology for eliciting a GAI network structure for the DM's preferences is an approach worth further research and consideration.

BIBLIOGRAPHY

FISHBURN, P.C. 1970. Utility theory for decision making. Wiley, New York.

KEENEY, R. 1976. Decisions with Multiple Objectives. Cambridge University Press, Cambridge.

BRAZIUNAS, D. AND BOUTILIER, C. 2005. Local Utility Elicitation in GAI Models. In Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), Anonymous AUAI Press, Arlington, Virginia, 42.

GONZALES, C. AND PERNY, P. 2004. GAI Networks for Utility Elicitation. In Principles of Knowledge Representation and Reasoning, Whistler, British Columbia, Canada.

KEENEY, R. 1996. Value-Focused Thinking. Harvard University Press, Cambridge.

BERNOULLI, D. 1954. Exposition of a New Theory on the Measurement of Risk. Econometrica 22.

VON NEUMANN, J. AND MORGANSTERN, O. 1944. Theory of Games and Economic Behavior. Princeton University Press, Princeton.

FISHBURN, P.C. 1981. Subjective expected utility: A review of normative theories. Theory and Decision.

HORVITZ, . 1988. Decision theory in expert systems and artificial intelligence. International journal of approximate reasoning.

BACCHUS, F. AND GROVE, A. 1995. Graphical Models for Preference and Utility. In Uncertainty in Artificial Intelligence. Proceedings of the Eleventh Conference , Anonymous Morgan Kaufmann Publishers, San Francisco.

IEEE STANDARD UPPER ONTOLOGY WORKING GROUP, 2005. Suggested Upper Merged Ontology (SUMO). http://ontology.teknowledge.com/.

NILES, I. AND PEASE, A. 2001. Towards a standard upper ontology. In FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems, Ogunquit, Maine, USA, Anonymous ACM Press, New York, NY, USA.

HANSSON, D.H. 2004. Ruby on Rails , www.rubyonrails.org .

GARRETT, J.J. 2005. Ajax: A New Approach to Web Applications. http://www.adaptivepath.com/ideas/essays/archives/000385.php.

HELSINKI UNIVERSITY OF TECHNOLOGY [SYSTEMS ANALYSIS LABORATORY]. 2005. Value Tree Analysis Theory. http://www.mcda.hut.fi/value_tree/theory/.

Appendix A. List of Value Trees from User Study Participants

User 1

I_{25}: Minimize commute time
- a_{79}: Close To Bus Terminal
- a_{80}: Close To College
- a_{78}: Close To Railway Terminal

I_{27}: Maximize the comfort of the indoor living space
- a_{7}: Has Bedroom
- a_{6}: Has Bathroom
- a_{11}: Has Kitchen
- a_{12}: Has Living Room
- a_{37}: Has Air Conditioning

I_{30}: Maximize proximity to culture
- a_{59}: Close To Movie Theatre
- a_{67}: Close To Pubs
- a_{82}: Close To Public Library
- a_{74}: Close To Beach

I_{32}: Maximize beauty of setting
- a_{5}: Has Balcony
- a_{55}: Has Mountain View
- a_{56}: Has Ocean View
- a_{76}: Close To Airport is false

I_{33}: Is near to and includes useful things
- a_{40}: Has Dishwasher
- a_{41}: Has Dryer
- a_{44}: Has Fridge
- a_{45}: Has Mailbox
- a_{48}: Has Shower
- a_{50}: Has Stove
- a_{52}: Has Toilet
- a_{53}: Has Washer
- a_{49}: Has Sink
- a_{112}: Has Laundry Facility
- a_{63}: Close To Grocery Store

I_{34}: Maximize comfort of bedroom
- a_{15}: Faces East
- a_{16}: Faces North
- a_{17}: Faces South
- a_{18}: Faces West
- a_{19}: Has Door
- a_{23}: Has Linoleum is false
- a_{24}: Has Vinyl is false
- a_{30}: Has Window Covering
- a_{34}: Has Large Area

I_{39}: Maximize ability to entertain

I_{38}: Maximize proximity to nature

I_{37}: Maximize energy efficiency

I_{36}: Maximize the comfort of the outdoor space

I_{35}: Maximize a healthy environment

I_{40}: Maximize safety

USER 2

I_{41}: Minimize commute time
- a_{79}: Close To Bus Terminal
- a_{89}: Close To College

I_{42}: Maximize a healthy environment
- a_{57}: Has Tree View
- a_{73}: Close To Park
- a_{74}: Close To Beach

I_{43}: Maximize the comfort of the indoor living space
- a_{112}: Has Laundry Facility
- a_{11}: Has Kitchen
- a_{7}: Has Bedroom
- a_{6}: Has Bathroom
- a_{5}: Has Balcony
- a_{12}: Has Living Room
- a_{44}: Has Fridge
- a_{49}: Has Sink
- a_{51}: Has Telephone Line
- a_{48}: Has Shower
- a_{52}: Has Toilet
- a_{46}: Has Oven
- a_{50}: Has Stove
- a_{57}: Has Tree View
- a_{53}: Has Washer
- a_{22}: Has Hardwood

I_{45}: Maximize energy efficiency
- a_{39}: Has Chimney is false

I_{48}: Maximize safety
- a_{39}: Has Chimney is false
- a_{47}: Has Security Alarm

I_{49}: Minimize costs
- a_{41}: Has Dryer is false
- a_{37}: Has Air Conditioning is false
- a_{33}: Has Medium Area

I_{50}: Maximize nearness to useful places (err)
- a_{63}: Close To Grocery Store
- a_{68}: Close To Restaurants
- a_{67}: Close To Pubs
- a_{102}: Close To Soccer Field
- a_{59}: Close To Movie Theatre

I_{51}: Maximize comfort of bedroom
- a_{19}: Has Door
- a_{22}: Has Hardwood
- a_{33}: Has Medium Area
- a_{16}: Faces North
- a_{30}: Has Window Covering

I_{52}: Maximize the comfort of the outdoor living space

I_{53}: Maximize proximity to nature

I_{54}: Maximize ability to entertain

User 3

I_{55}: Minimize commute time

- a_{79}: Close To Bus Terminal
- a_{63}: Close To Grocery Store
- a_{68}: Close To Restaurants

I_{56}: Maximize a healthy environment

- a_{74}: Close To Beach
- a_{73}: Close To Park
- a_{75}: Close To Tourist Attraction is false

I_{57}: Maximize the comfort of the indoor living space

- a_{13}: Has Office
- a_{22}: Has Hardwood
- a_{6}: Has Bathroom
- a_{7}: Has Bedroom
- a_{11}: Has Kitchen
- a_{9}: Has Foyer
- a_{23}: Has Linoleum is false
- a_{24}: Has Vinyl is false
- a_{29}: Has Wallpaper is false
- a_{34}: Has Large Area
- a_{41}: Has Dryer
- a_{44}: Has Fridge
- a_{48}: Has Shower
- a_{50}: Has Stove
- a_{52}: Has Toilet
- a_{53}: Has Washer
- a_{46}: Has Oven

I_{58}: Maximize the comfort of the outdoor living space

- a_{5}: Has Balcony

I_{59}: Maximize energy efficiency

- a_{79}: Close To Bus Terminal

I_{61}: Maximize ability to entertain

- a_{16}: Faces North
- a_{55}: Has Mountain View
- a_{56}: Has Ocean View
- a_{69}: Close To Movie Theatre

I_{62}: Maximize safety

- a_{70}: Close To Major Road Intersection is false
- a_{47}: Has Security Alarm
- a_{115}: Has Parking Garage

I_{63}: Maximize proximity to nature

USER 4

- I_{64}: Minimize commute time
  - a_{59}: Close To Movie Theatre
  - a_{67}: Close To Pubs
  - a_{63}: Close To Grocery Store
  - a_{68}: Close To Restaurants
  - a_{74}: Close To Beach
  - a_{73}: Close To Park
  - a_{79}: Close To Bus Terminal
  - a_{89}: Close To College
  - a_{107}: Close To Swimming Pool
  - a_{108}: Close To Gymnasium
  - a_{106}: Close To Bank
- I_{65}: Maximize a healthy environment
- I_{69}: Maximize proximity to nature
  - a_{56}: Has Ocean View
- I_{70}: Maximize ability to entertain
- I_{71}: Maximize safety
  - a_{47}: Has Security Alarm
- I_{72}: Maximize comfort of bedroom
  - a_{21}: Has Carpet
  - a_{17}: Faces South
- I_{73}: maximize the facilities in kitchen
  - a_{26}: Has Oil Paint
  - a_{39}: Has Chimney
  - a_{40}: Has Dishwasher
  - a_{44}: Has Fridge
  - a_{46}: Has Oven
  - a_{50}: Has Stove
  - a_{53}: Has Washer
  - a_{34}: Has Large Area
  - a_{49}: Has Sink
- I_{75}: maximize the comfort in living room
  - a_{26}: Has Oil Paint
  - a_{43}: Has Fireplace
  - a_{51}: Has Telephone Line
  - a_{5}: Has Balcony
  - a_{19}: Has Door
  - a_{21}: Has Carpet
  - a_{38}: Has Chair
- I_{77}: maximize good neighborhood
  - a_{115}: Has Parking Garage
  - a_{116}: Has Parking Lot
- I_{78}: maximize the other useful feature in the apartment
  - a_{118}: Has Swimming Pool
  - a_{45}: Has Mailbox
- I_{79}: maximize the features in the apartment
  - a_{5}: Has Balcony
  - a_{7}: Has Bedroom
  - a_{11}: Has Kitchen
  - a_{12}: Has Living Room
  - a_{13}: Has Office
  - a_{34}: Has Large Area
  - a_{48}: Has Shower
  - a_{52}: Has Toilet
- I_{80}: Maximize the comfort of the indoor living space
- I_{81}: Maximize the comfort of the outdoor living space
- I_{82}: Maximize energy efficiency

User 5

I_{91}: Minimize commute time
- a_{89}: Close To College

I_{97}: Maximize ability to entertain
- a_{12}: Has Living Room
- a_{33}: Has Medium Area
- a_{34}: Has Large Area

I_{98}: Maximize safety

I_{99}: Maximize proximity to a good gym
- a_{108}: Close To Gymnasium

I_{100}: Maximize proximity to a park with a track
- a_{73}: Close To Park
- a_{85}: Close To High School
- a_{102}: Close To Soccer Field
- a_{103}: Close To Baseball Field

I_{101}: Maximize proximity to cheap produce stores
- a_{63}: Close To Grocery Store

I_{102}: Maximize proximity to a big grocery store
- a_{63}: Close To Grocery Store
- a_{65}: Close To Wholesale Store

I_{104}: Is in an urban area with services and things to
- a_{59}: Close To Movie Theatre
- a_{67}: Close To Pubs
- a_{68}: Close To Restaurants
- a_{72}: Close To Laundromat
- a_{74}: Close To Beach

I_{105}: Maximize proximity to a bus route
- a_{70}: Close To Major Road Intersection

I_{106}: Big kitchen
- a_{11}: Has Kitchen

I_{107}: Strong water pressure

I_{108}: Room for secure bike storage

I_{109}: Is above ground

I_{110}: Is relatively new

I_{111}: Hardwood floors
- a_{22}: Has Hardwood

I_{112}: Well insulated

I_{113}: Access to climate control

I_{114}: Doesn't require too many major purchases to make
- a_{90}: Has Stove
- a_{44}: Has Fridge

I_{115}: Maximize a healthy environment

I_{116}: Maximize the comfort of the indoor living space

I_{117}: Maximize the comfort of the outdoor living space

I_{118}: Maximize energy efficiency

I_{119}: Maximize proximity to nature

User 6

I_{120}: Minimize commute time
- a_{70}: Close To Major Road Intersection
- a_{89}: Close To College
- a_{79}: Close To Bus Terminal

I_{122}: Maximize the comfort of the indoor living space
- a_{34}: Has Large Area
- a_{37}: Has Air Conditioning
- a_{22}: Has Hardwood
- a_{19}: Has Door
- a_{11}: Has Kitchen
- a_{12}: Has Living Room

I_{126}: Maximize ability to entertain
- a_{51}: Has Telephone Line
- a_{59}: Close To Movie Theatre
- a_{67}: Close To Pubs
- a_{73}: Close To Park
- a_{102}: Close To Soccer Field

I_{128}: Maximize a healthy environment

I_{129}: Maximize the comfort of the outdoor living space

I_{130}: Maximize energy efficiency

I_{131}: Maximize proximity to nature

I_{132}: Maximize safety

USER 7

I_{133}: Minimize commute time
- a_{89}: Close To College
- a_{70}: Close To Major Road Intersection
- a_{63}: Close To Grocery Store
- a_{66}: Close To Shopping Mall

I_{134}: Maximize convenience
- a_{40}: Has Dishwasher
- a_{53}: Has Washer
- a_{41}: Has Dryer
- a_{62}: Close To Drug Store
- a_{70}: Close To Major Road Intersection
- a_{71}: Close To Parking Lot
- a_{112}: Has Laundry Facility
- a_{115}: Has Parking Garage
- a_{116}: Has Parking Lot

I_{135}: Satisfies essentials of indoor living
- a_{11}: Has Kitchen
- a_{5}: Has Balcony
- a_{7}: Has Bedroom
- a_{12}: Has Living Room
- a_{8}: Has Den
- a_{21}: Has Carpet
- a_{33}: Has Medium Area
- a_{46}: Has Oven
- a_{42}: Has Fan
- a_{51}: Has Telephone Line
- a_{32}: Has Small Area is false
- a_{30}: Has Window Covering
- a_{44}: Has Fridge
- a_{48}: Has Shower
- a_{50}: Has Stove
- a_{52}: Has Toilet
- a_{49}: Has Sink

I_{136}: Satisfies Comfortable living
- a_{37}: Has Air Conditioning
- a_{55}: Has Mountain View
- a_{43}: Has Fireplace
- a_{56}: Has Ocean View
- a_{57}: Has Tree View
- a_{17}: Faces South
- a_{15}: Faces East
- a_{68}: Close To Restaurants
- a_{92}: Close To Public Library

I_{139}: Maximize ability to entertain
- a_{5}: Has Balcony
- a_{12}: Has Living Room
- a_{21}: Has Carpet
- a_{22}: Has Hardwood
- a_{40}: Has Dishwasher
- a_{46}: Has Oven
- a_{42}: Has Fan
- a_{55}: Has Mountain View
- a_{70}: Close To Major Road Intersection
- a_{71}: Close To Parking Lot
- a_{19}: Has Door
- a_{28}: Has Painting

I_{141}: Allow space for work
- a_{13}: Has Office
- a_{8}: Has Den

I_{142}: Maximize ease of maintenance
- a_{29}: Has Wallpaper
- a_{22}: Has Hardwood is false

I_{143}: Maximize Healthy Living
- a_{108}: Close To Gymnasium
- a_{107}: Close To Swimming Pool
- a_{73}: Close To Park
- a_{74}: Close To Beach
- a_{111}: Has Grounds
- a_{117}: Has Gymnasium
- a_{118}: Has Swimming Pool

I_{144}: Maximize a healthy environment

I_{145}: Maximize the comfort of the indoor living space

I_{146}: Maximize the comfort of the outdoor living space

I_{147}: Maximize energy efficiency

I_{148}: Maximize proximity to nature

I_{149}: Maximize safety

USER 8

I_{150}: Minimize travel requirements
- a_{59}: Close To Movie Theatre
- a_{60}: Close To Live Theatre
- a_{62}: Close To Drug Store
- a_{63}: Close To Grocery Store
- a_{67}: Close To Pubs
- a_{68}: Close To Restaurants
- a_{82}: Close To Public Library
- a_{74}: Close To Beach
- a_{73}: Close To Park
- a_{89}: Close To College
- a_{106}: Close To Bank

I_{152}: Maximize the comfort of the indoor living space
- a_7: Has Bedroom
- a_{11}: Has Kitchen
- a_6: Has Bathroom
- a_{12}: Has Living Room
- a_{13}: Has Office
- a_{23}: Has Linoleum
- a_{21}: Has Carpet
- a_{30}: Has Window Covering
- a_{34}: Has Large Area
- a_{37}: Has Air Conditioning
- a_{19}: Has Door
- a_{70}: Close To Major Road Intersection is false

I_{153}: Maximize the comfort of the outdoor living space
- a_5: Has Balcony
- a_{55}: Has Mountain View
- a_{56}: Has Ocean View
- a_{57}: Has Tree View
- a_{71}: Close To Parking Lot is false
- a_{111}: Has Grounds

I_{154}: Maximize energy efficiency
- a_{30}: Has Window Covering

I_{157}: Maximize safety
- a_{19}: Has Door

I_{158}: Maximize serviceability
- a_{41}: Has Dryer
- a_{40}: Has Dishwasher
- a_{44}: Has Fridge
- a_{46}: Has Oven
- a_{49}: Has Sink
- a_{48}: Has Shower
- a_{51}: Has Telephone Line
- a_{52}: Has Toilet
- a_{53}: Has Washer
- a_{45}: Has Mailbox
- a_{112}: Has Laundry Facility

I_{159}: Minimize commute time

I_{160}: Maximize a healthy environment

I_{161}: Maximize proximity to nature

I_{162}: Maximize ability to entertain

User 9

I_{163}: Minimize commute time
- a_{63}: Close To Grocery Store
- a_{65}: Close To Wholesale Store
- a_{66}: Close To Shopping Mall
- a_{79}: Close To Bus Terminal
- a_{106}: Close To Bank

I_{164}: Maximize a healthy environment
- a_{74}: Close To Beach
- a_{73}: Close To Park
- a_{56}: Has Ocean View

I_{165}: Maximize the comfort of the indoor living space
- a_5: Has Balcony
- a_6: Has Bathroom
- a_7: Has Bedroom
- a_{11}: Has Kitchen
- a_{12}: Has Living Room
- a_{17}: Faces South
- a_{22}: Has Hardwood
- a_{29}: Has Wallpaper
- a_{34}: Has Large Area
- a_{40}: Has Dishwasher
- a_{41}: Has Dryer
- a_{46}: Has Oven
- a_{48}: Has Shower
- a_{49}: Has Sink
- a_{50}: Has Stove
- a_{51}: Has Telephone Line
- a_{52}: Has Toilet

I_{166}: Maximize the comfort of the outdoor living space
- a_{74}: Close To Beach
- a_{73}: Close To Park
- a_{36}: Has Aerobic Exercise Device

I_{170}: Maximize safety
- a_{47}: Has Security Alarm

I_{171}: Maximize energy efficiency

I_{172}: Maximize proximity to nature

I_{173}: Maximize ability to entertain

USER 10

I_9: Minimize commute time
- a_{89}: Close To College
- a_{79}: Close To Bus Terminal
- a_{116}: Has Parking Lot

I_{10}: Maximize proximity to nature
- a_{57}: Has Tree View
- a_{73}: Close To Park
- a_{74}: Close To Beach
- a_{115}: Has Parking Garage

I_{11}: Maximize the comfort of the indoor living space
- a_6: Has Bathroom
- a_7: Has Bedroom
- a_{11}: Has Kitchen
- a_{12}: Has Living Room
- a_{16}: Faces North
- a_{17}: Faces South
- a_{21}: Has Carpet
- a_{30}: Has Window Covering
- a_{33}: Has Medium Area
- a_{39}: Has Chimney
- a_{40}: Has Dishwasher
- a_{41}: Has Dryer
- a_{43}: Has Fireplace
- a_{44}: Has Fridge
- a_{45}: Has Mailbox
- a_{46}: Has Oven
- a_{48}: Has Shower
- a_{49}: Has Sink
- a_{50}: Has Stove
- a_{52}: Has Toilet
- a_{53}: Has Washer
- a_{57}: Has Tree View

I_{13}: Maximize energy efficiency
- a_{17}: Faces South
- a_{19}: Has Door
- a_{74}: Close To Beach

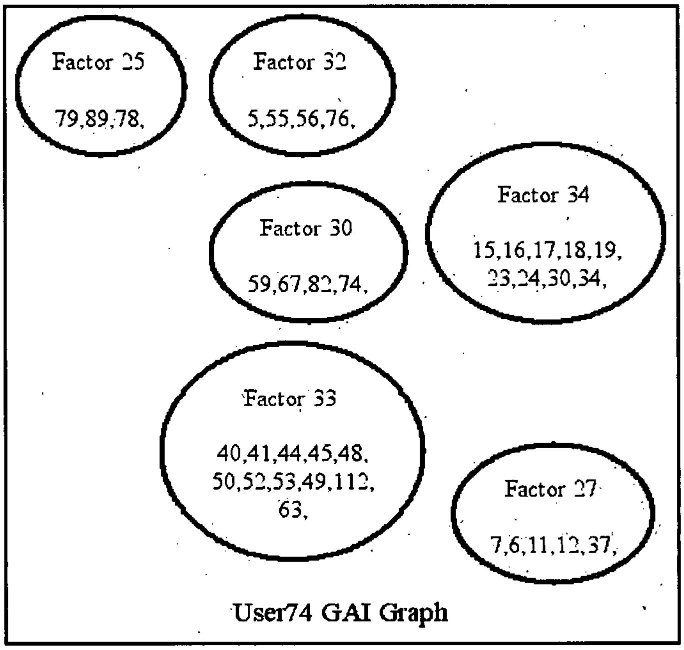
I_{14}: Maximize safety
- a_5: Has Balcony is false
- a_{19}: Has Door
- a_{71}: Close To Parking Lot is false
- a_{116}: Has Parking Lot is false

I_{15}: Maximize comfort of guests
- a_{19}: Has Door
- a_{34}: Has Large Area
- a_{43}: Has Fireplace
- a_{52}: Has Toilet
- a_{76}: Close To Airport
- a_{79}: Close To Bus Terminal
- a_{116}: Has Parking Lot

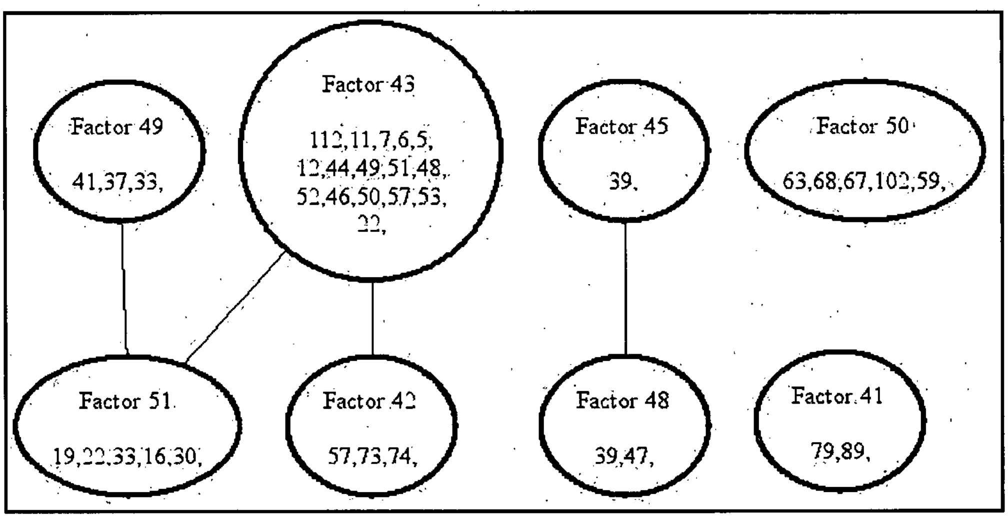
I_{16}: Easy access to entertainment
- a_{59}: Close To Movie Theatre
- a_{73}: Close To Park
- a_{75}: Close To Tourist Attraction
- a_{84}: Close To Museum

Appendix B. LIST OF GAI NETWORKS GENERATED FROM USER STUDY PARTICIPANTS

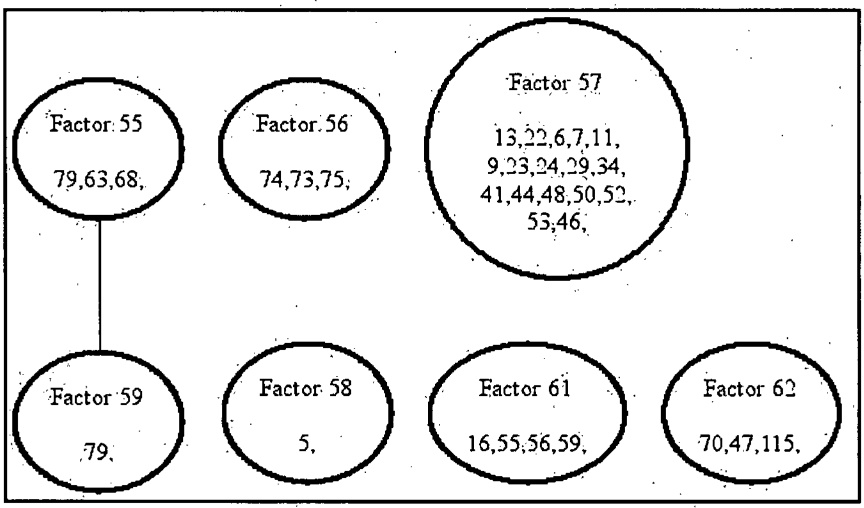
USER 1



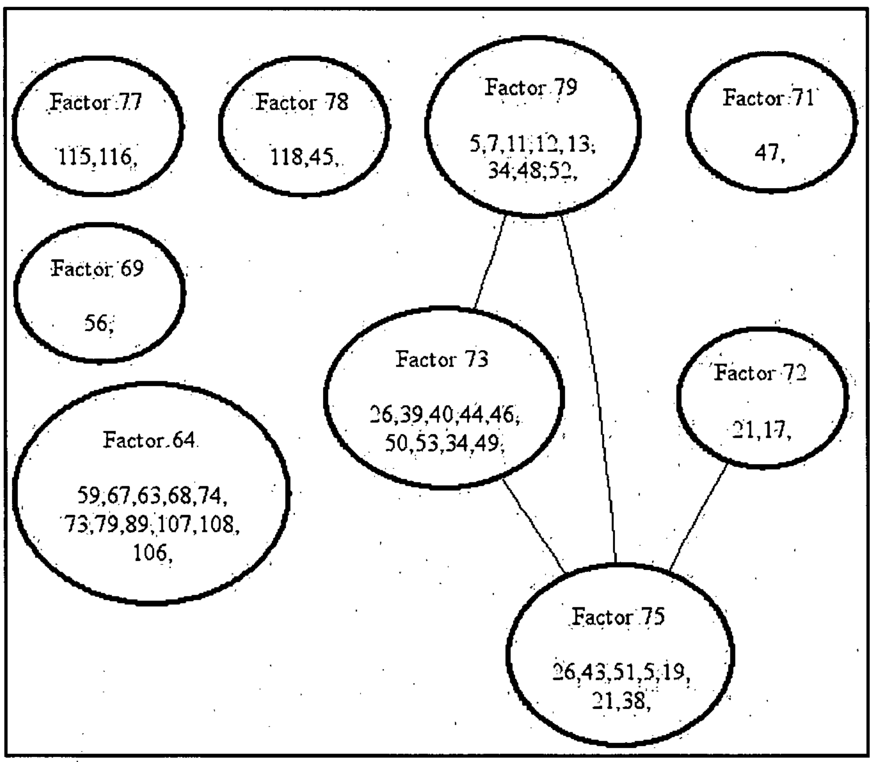
USER 2



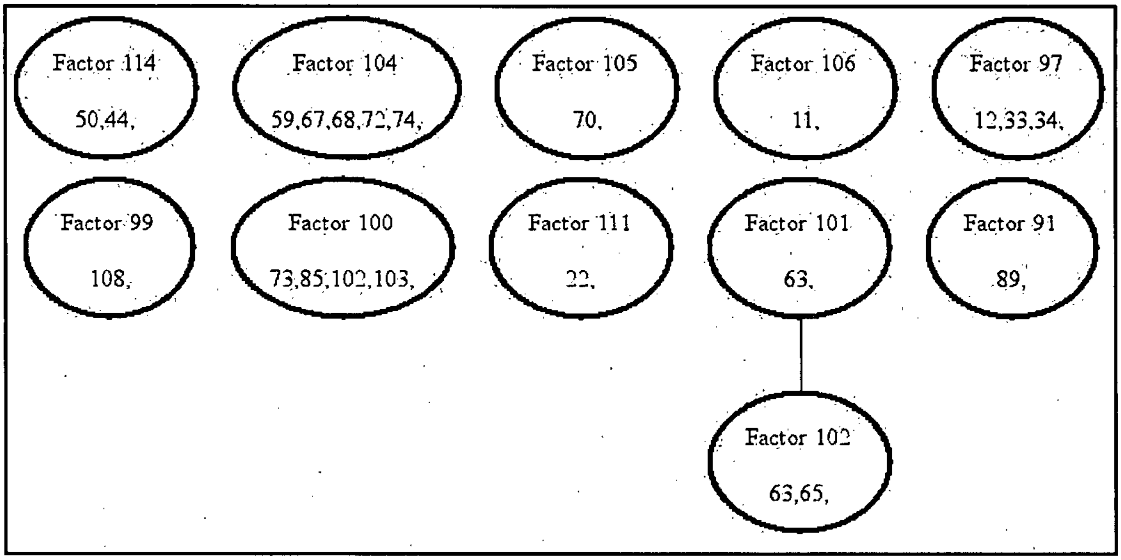
USER 3


Factor 55
79,63,68,
Factor 56
74,73,75,
Factor 57
13,22,6,7,11,
9,23,24,29,34,
41,44,48,50,52,
53,46,
Factor 59
79,
Factor 58
5,
Factor 61
16,55,56,59,
Factor 62
70,47,115,

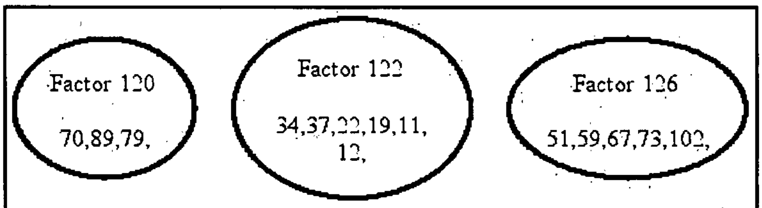
USER 4


Factor 77
115,116,
Factor 78
118,45,
Factor 79
5,7,11,12,13,
34,48,52,
Factor 71
47,
Factor 69
56,
Factor 73
26,39,40,44,46,
50,53,34,49,
Factor 72
21,17,
Factor 64
59,67,63,68,74,
73,79,89,107,108,
106,
Factor 75
26,43,51,5,19,
21,38,

USER 5

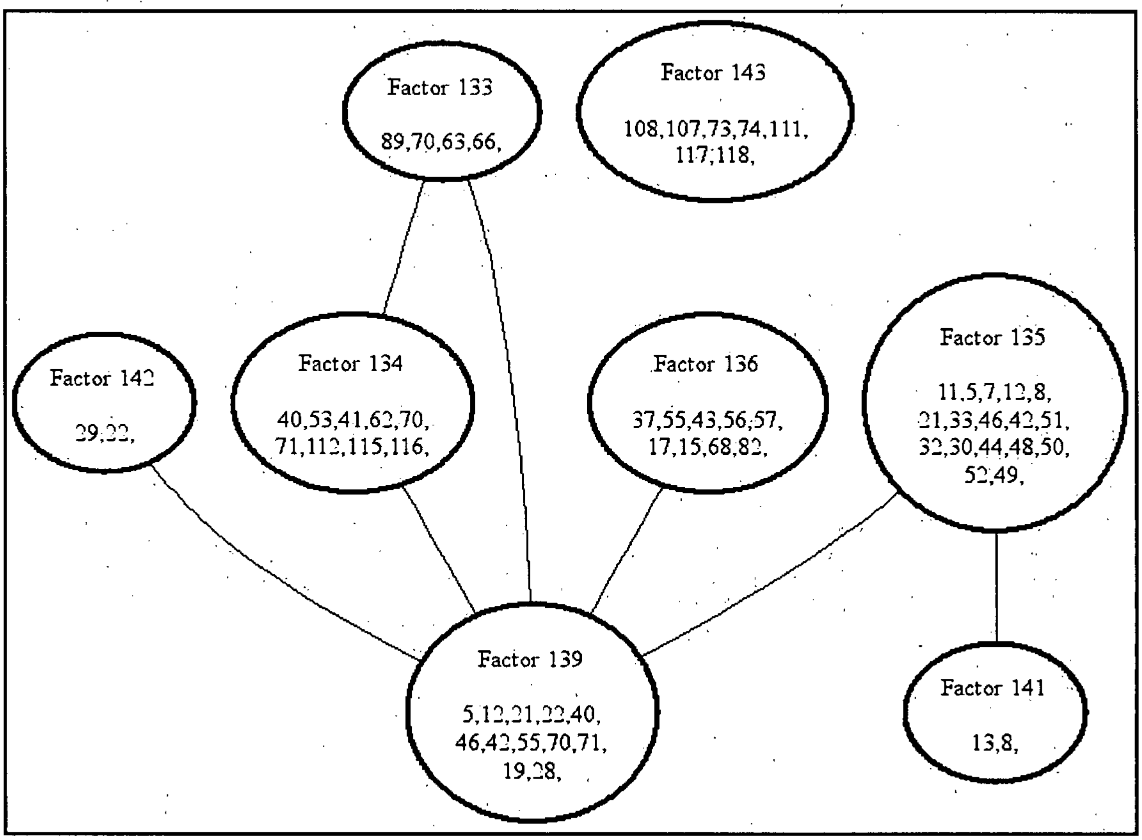


USER 6

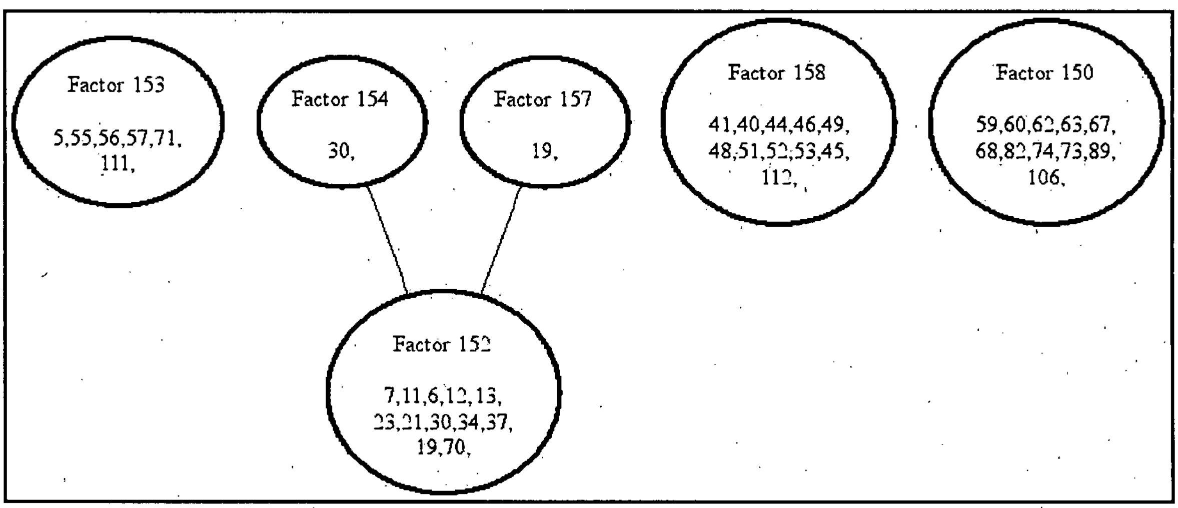


User805 GAI Graph

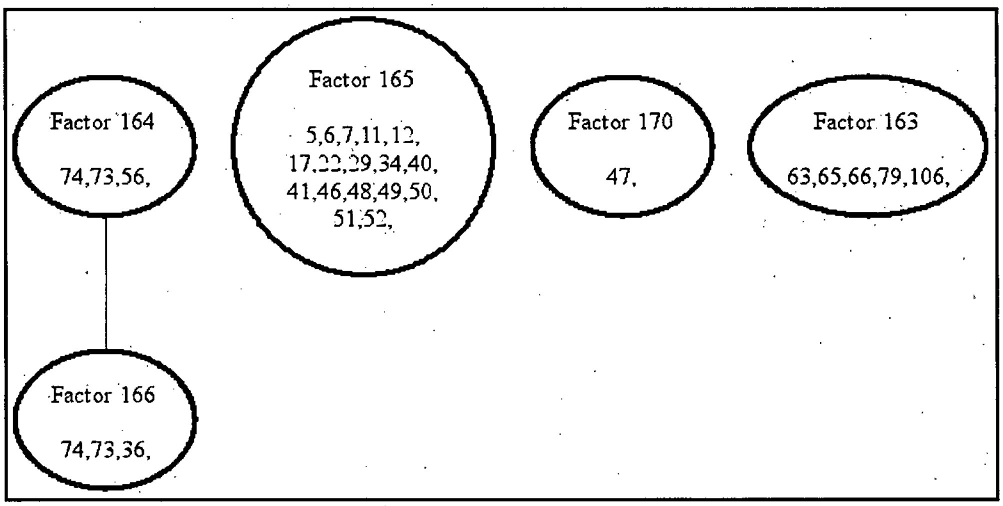
USER 7


Factor 133
89,70,63,66,
Factor 143
108,107,73,74,111,
117,118,
Factor 142
29,22,
Factor 134
40,53,41,62,70,
71,112,115,116,
Factor 136
37,55,43,56,57,
17,15,68,82,
Factor 135
11,5,7,12,8,
21,33,46,42,51,
32,30,44,48,50,
52,49,
Factor 139
5,12,21,22,40,
46,42,55,70,71,
19,28,
Factor 141
13,8,

USER 8

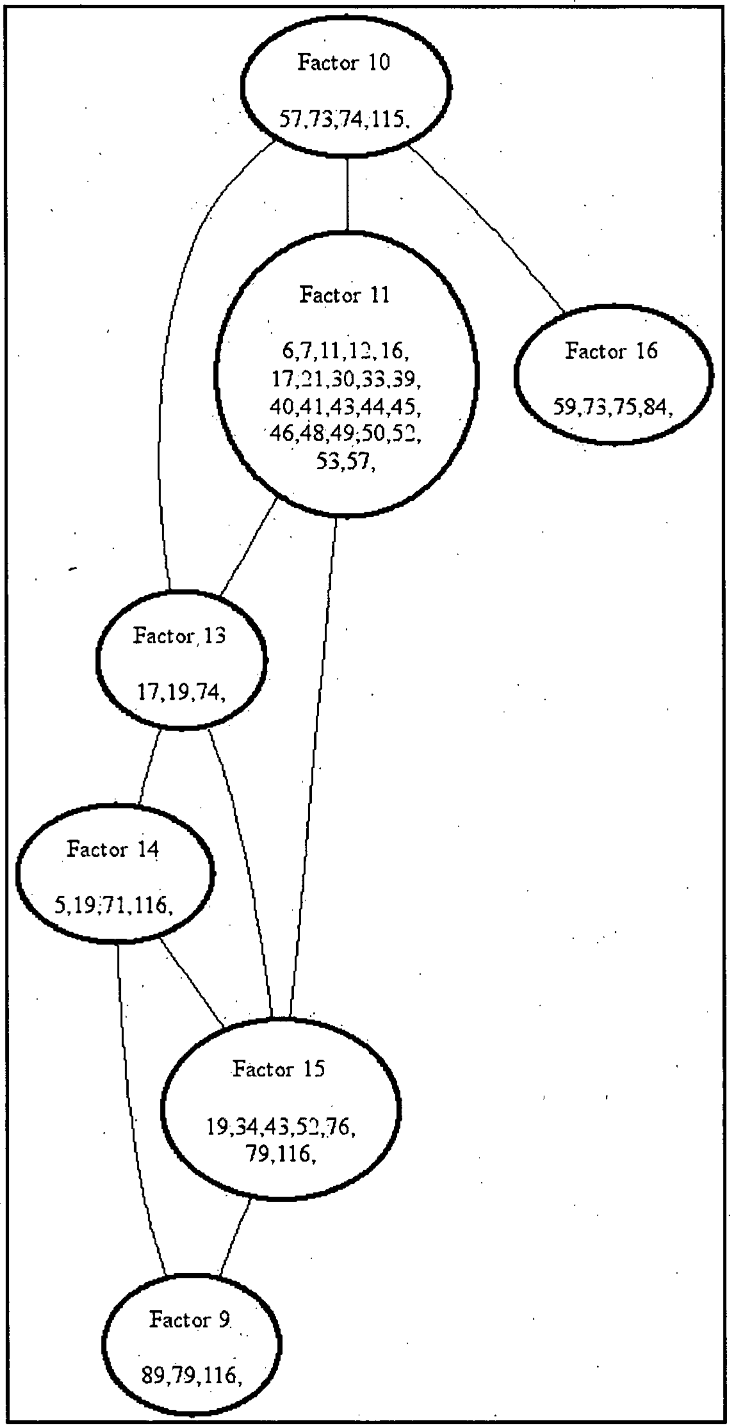

Factor 153
5,55,56,57,71,
111,
Factor 154
30,
Factor 157
19,
Factor 158
41,40,44,46,49,
48,51,52,53,45,
112,
Factor 150
59,60,62,63,67,
68,82,74,73,89,
106,
Factor 152
7,11,6,12,13,
23,21,30,34,37,
19,70,

USER 9



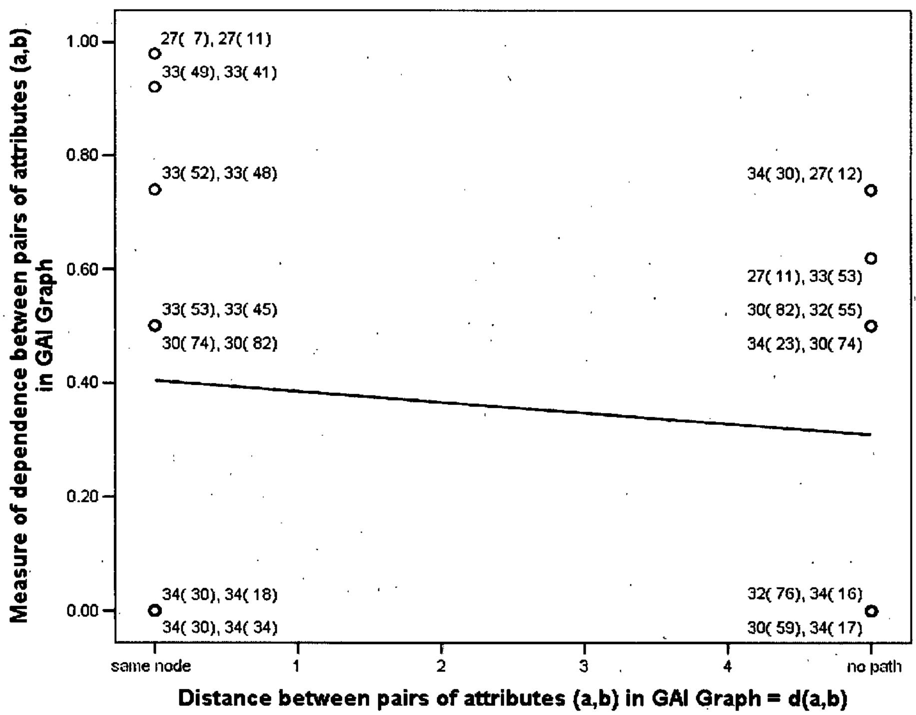
Factor 164
74,73,56,
Factor 165
5,6,7,11,12,
17,22,29,34,40,
41,46,48,49,50,
51,52,
Factor 170
47,
Factor 163
63,65,66,79,106,
Factor 166
74,73,36,

USER 10

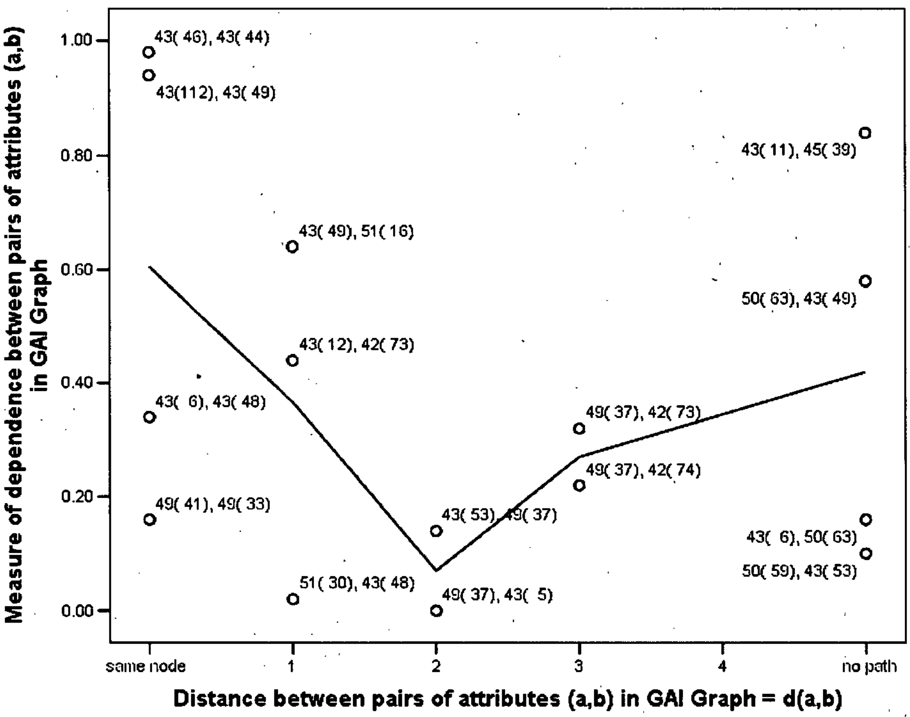

Factor 10
57,73,74,115,
Factor 11
6,7,11,12,16,
17,21,30,33,39,
40,41,43,44,45,
46,48,49,50,52,
53,57,
Factor 16
59,73,75,84,
Factor 13
17,19,74,
Factor 14
5,19,71,116,
Factor 15
19,34,43,52,76,
79,116,
Factor 9
89,79,116,

Appendix C. LIST OF SCATTER PLOTS OF MEASURED DEPENDENCIES BETWEEN PAIRS OF ATTRIBUTES AND DISTANCE BETWEEN PAIRS OF ATTRIBUTES OF USER STUDY PARTICIPANTS

USER 1

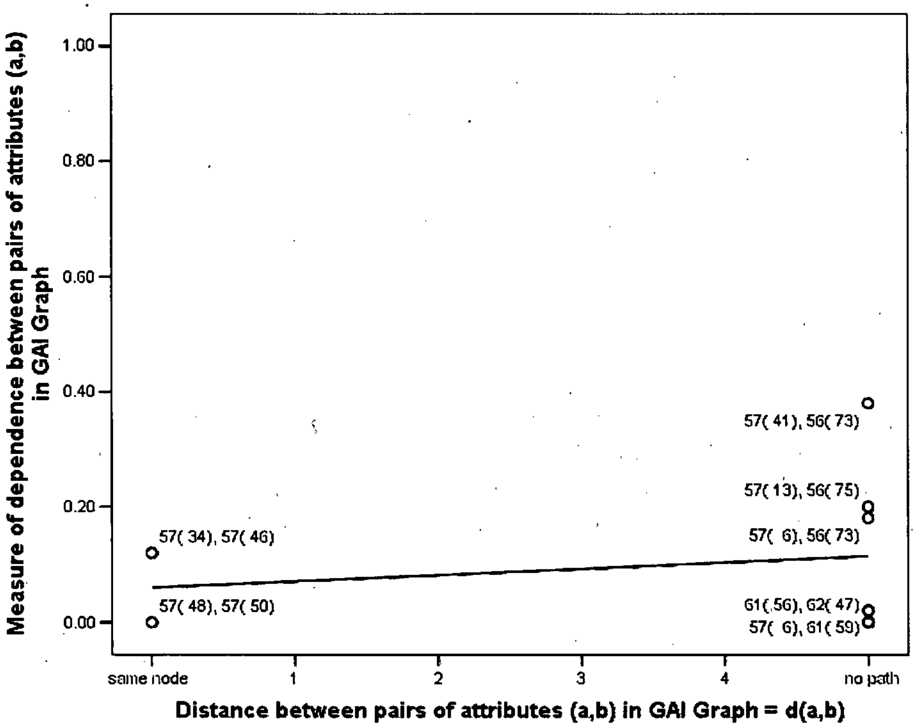


USER 2



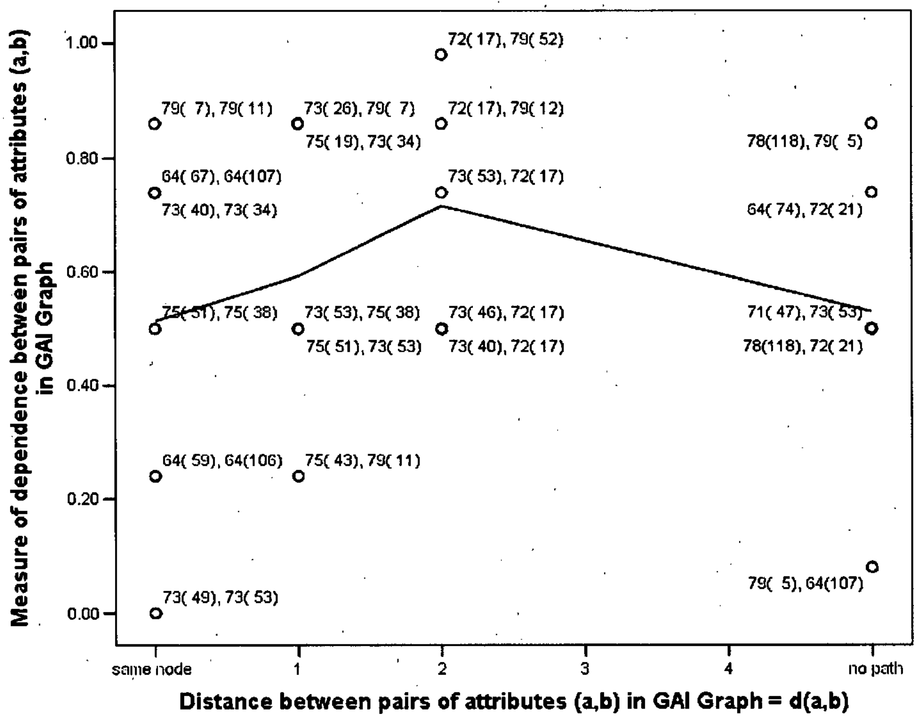
Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
43(46), 43(44)
43(112), 43(49)
43(11), 45(39)
43(49), 51(16)
50(63), 43(49)
43(12), 42(73)
43(6), 43(48)
49(37), 42(73)
49(37), 42(74)
49(41), 49(33)
43(53), 49(37)
43(6), 50(63)
50(59), 43(53)
51(30), 43(48)
49(37), 43(5)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

USER 3



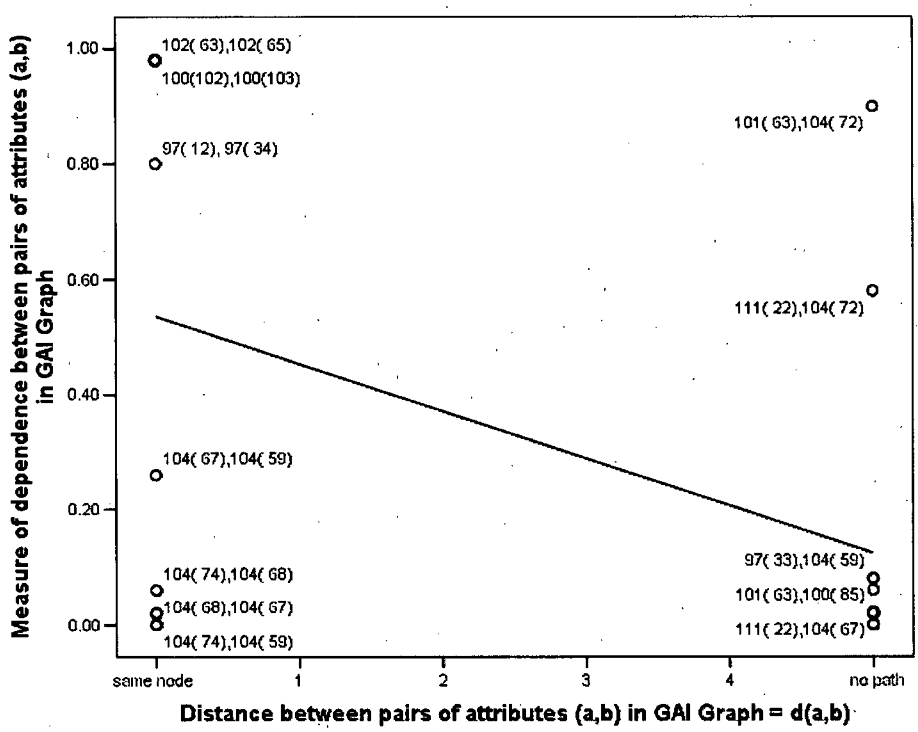
Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
57(41), 56(73)
57(13), 56(75)
57(34), 57(46)
57(6), 56(73)
57(48), 57(50)
61(56), 62(47)
57(6), 61(59)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

USER 4



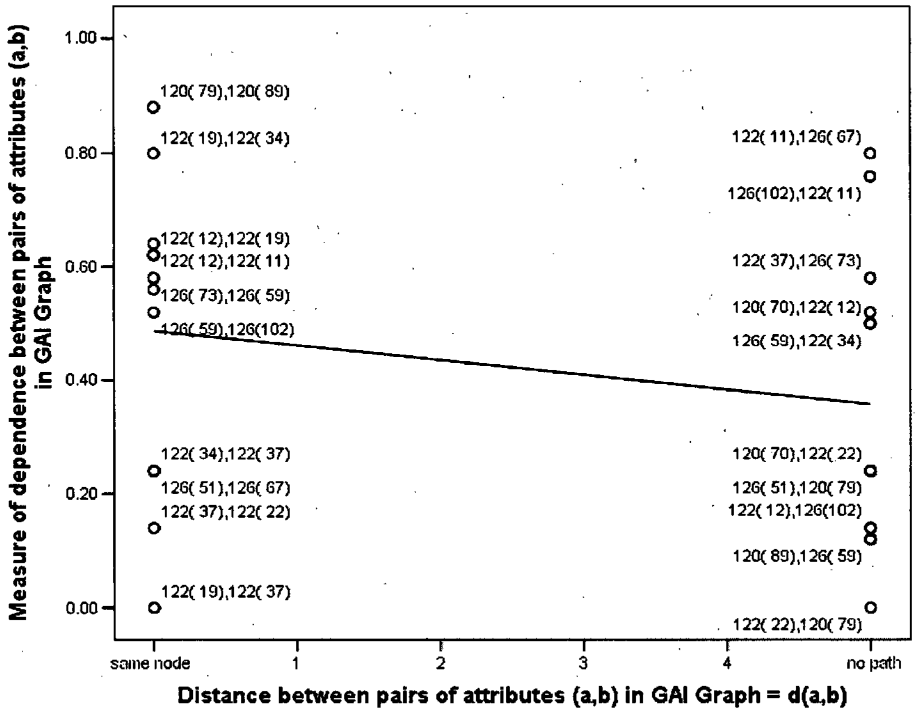
Measure of dependence between pairs of attributes (a,b) in GAI Graph
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)
72(17), 79(52)
79(7), 79(11)
73(26), 79(7)
75(19), 73(34)
72(17), 79(12)
78(118), 79(5)
64(67), 64(107)
73(40), 73(34)
73(53), 72(17)
64(74), 72(21)
75(51), 75(38)
73(53), 75(38)
75(51), 73(53)
73(46), 72(17)
73(40), 72(17)
71(47), 73(53)
78(118), 72(21)
64(59), 64(106)
75(43), 79(11)
79(5), 64(107)
73(49), 73(53)
same node
no path

USER 5



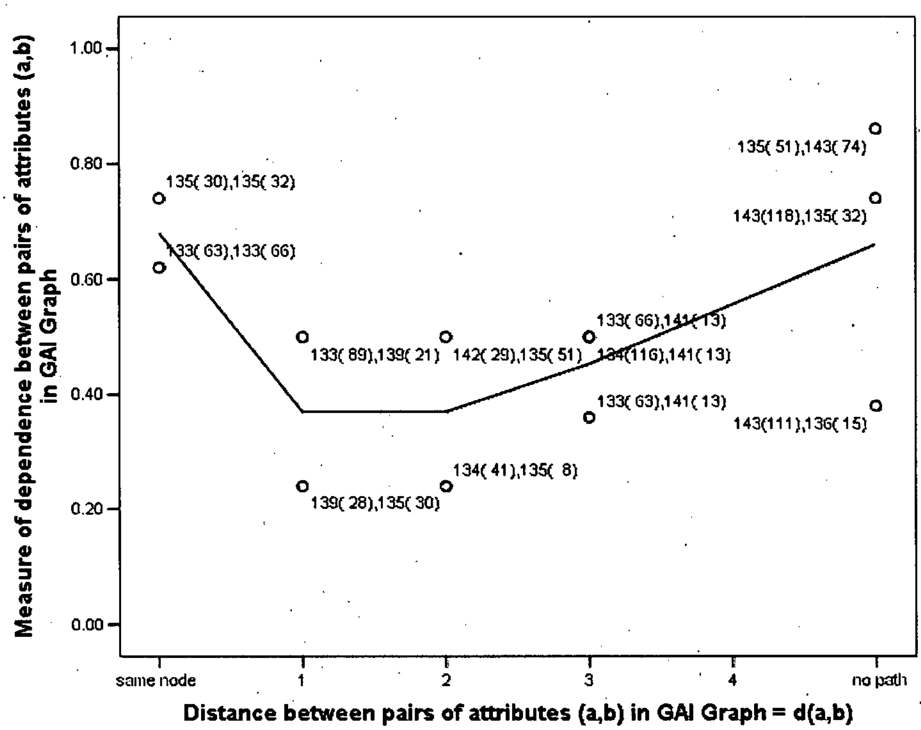
Measure of dependence between pairs of attributes (a,b) in GAI Graph
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)
102(63),102(65)
100(102),100(103)
101(63),104(72)
97(12), 97(34)
111(22),104(72)
104(67),104(59)
104(74),104(68)
104(68),104(67)
104(74),104(59)
97(33),104(59)
101(63),100(85)
111(22),104(67)
same node
no path

USER 6



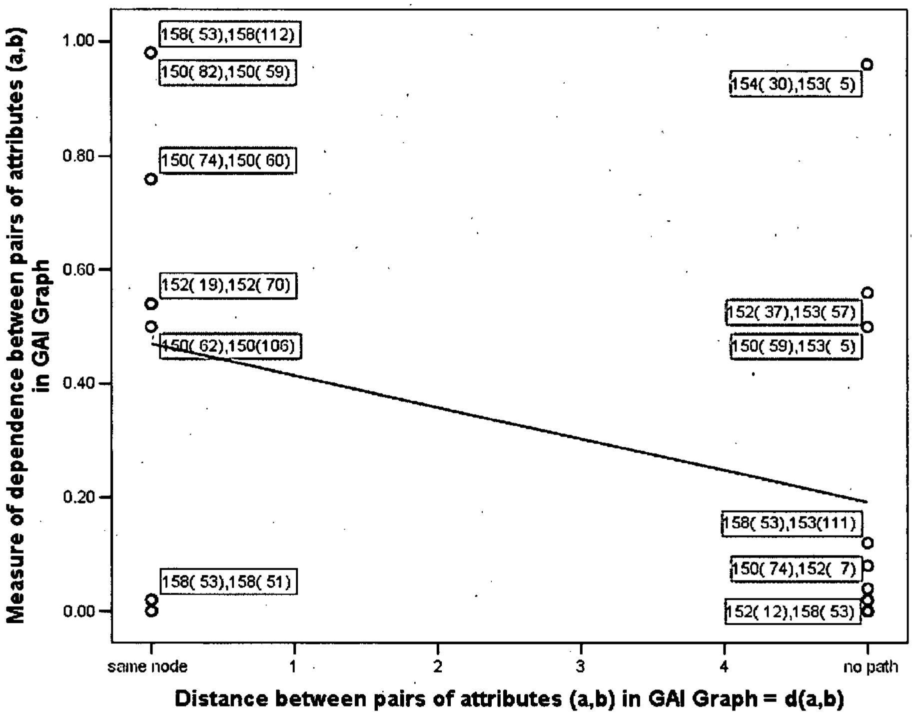
Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
120(79),120(89)
122(19),122(34)
122(12),122(19)
122(12),122(11)
126(73),126(59)
126(59),126(102)
122(34),122(37)
126(51),126(67)
122(37),122(22)
122(19),122(37)
122(11),126(67)
126(102),122(11)
122(37),126(73)
120(70),122(12)
126(59),122(34)
120(70),122(22)
126(51),120(79)
122(12),126(102)
120(89),126(59)
122(22),120(79)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

USER 7

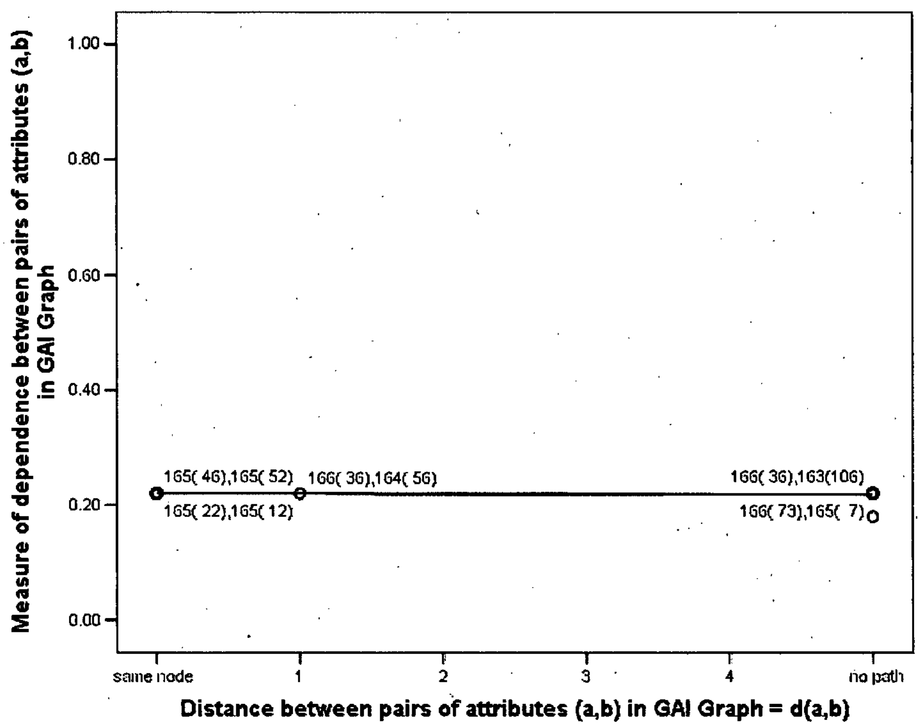


Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
135(30),135(32)
133(63),133(66)
133(89),139(21)
142(29),135(51)
133(66),141(13)
134(116),141(13)
133(63),141(13)
134(41),135(8)
139(28),135(30)
135(51),143(74)
143(118),135(32)
143(111),136(15)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

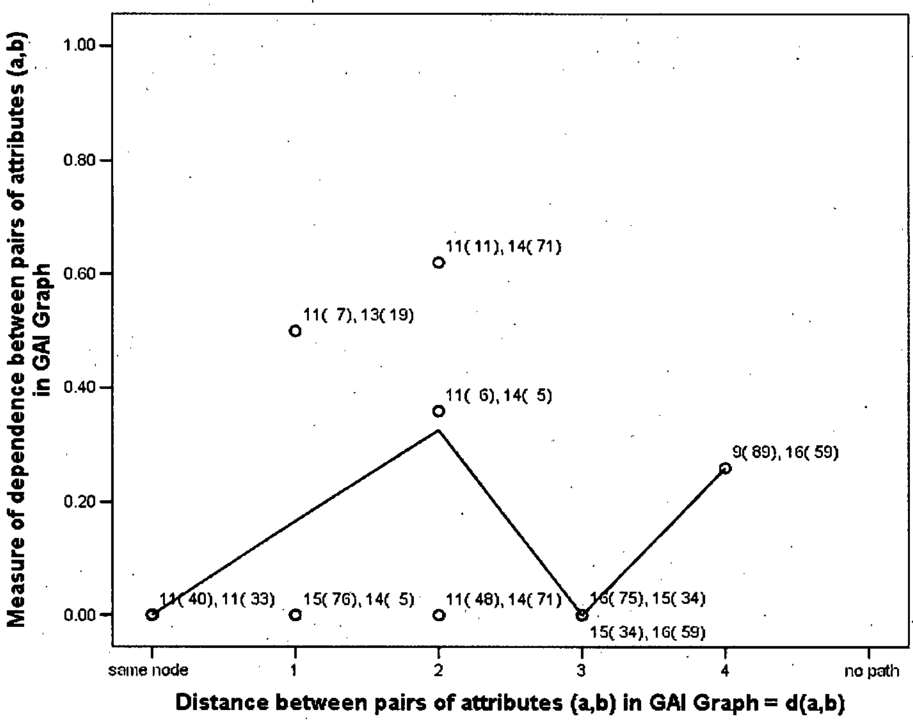
USER 8


Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
158(53),158(112)
150(82),150(59)
150(74),150(60)
152(19),152(70)
150(62),150(106)
158(53),158(51)
154(30),153(5)
152(37),153(57)
150(59),153(5)
158(53),153(111)
150(74),152(7)
152(12),158(53)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

USER 9


Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
165(46),165(52)
166(36),164(56)
166(36),163(106)
165(22),165(12)
166(73),165(7)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

USER 10



Measure of dependence between pairs of attributes (a,b) in GAI Graph
1.00
0.80
0.60
0.40
0.20
0.00
11(11), 14(71)
11(7), 13(19)
11(6), 14(5)
9(89), 16(59)
11(40), 11(33)
15(76), 14(5)
11(48), 14(71)
16(75), 15(34)
15(34), 16(59)
same node
1
2
3
4
no path
Distance between pairs of attributes (a,b) in GAI Graph = d(a,b)

Appendix D. LIST OF COMMENTS/SUGGESTIONS/QUESTIONS COMPILED FROM USER STUDY PARTICIPANTS

DEFAULT VALUE TREE

- Quite clear and intuitive (User 2)
- The tree covers pretty much what I need when I search for an apartment (User 9)
- Well Selected Objectives for the Value Tree (User 5)
- Good Objectives but a strong tendency to use the default value tree (User 1)
- Perhaps provide verbal suggestions instead (User 1)
- Perhaps a couple more default categories to get me thinking (too few makes me have to think about groupings) (User 6)
- I did not use many of the default Value tree concepts. I wasn't thinking in terms of maximization as much as fulfilling different requirements of like the apartment will be used for parties, work, relaxation, cooking etc. (User 7)

ONTOLOGY

- Clear and intuitive (User 2)
- Needs more options and be broader (User 9)
- Odd to have to add all the leaves when really, suppose, all I wanted was a window (not caring which way it faced) (User 1)
- Do changes in the ontology affect a user's value tree? (User 2)
- Maybe the ontology should have other types of attributes besides boolean (i.e. true and false) such as quantity or quality / thresholds (User 3)
- Felt incomplete, such as no "bus stop" just "bus terminal" etc (User 5)
- Some concepts such as doors and sinks are essential yet others are optional such as "close to park"; difficult to combine both. (User 7)

INTERFACE

- Easy to use interface (User 2, User 7)
- Slick (User 3)
- Good (User 5)
- Improve slow updating (User 2)
- It can be improved, quite slow updates (User 4)
- Perhaps have the tree not expanded/condensed by default, with smaller font sizes for "less" important attributes. (User 6, User 10)
- Suggest collapsing ontology to make navigation easier (User 5)

OTHER

- Structure of questions during validation was difficult. Would there be any other way of posing them? (User 1)
- Hard part is answering comparison gamble queries especially when one item is mandatory and one is superfluous such as 'sink' and 'near a park' (User 7)

Appendix E. EMAIL FOR PRE-STUDY

...

Suppose you were searching for an apartment. You want the place that you are searching for to fulfill certain goals and objectives you have, such as, you want to reduce the amount of time it takes for you to commute to either work/school from home - so you want to "minimize commute time".

I want to compile a list of high-level objectives that are most common amongst people. If you could please take a look at the following list and let me know whether

- something should be removed (which number)
- something should be added (which number)
- the wording needs to be changed so that it is clear immediately what is being talked about (which number)

and just provide a reason for it (you can have any sort of reasoning behind your decision - after all your choices are personal). The goal for this is to find out whether I've been able to capture the majority of things people care about when searching for a new place and if I have missed anything MAJOR!

This is the list of fundamental objectives:

- Minimize commute time
- Maximize a healthy environment
- Maximize the comfort of the indoor living space
- Maximize the comfort of the outdoor living space
- Maximize energy efficiency
- Maximize proximity to nature
- Maximize ability to entertain
- Maximize safety

Appendix F. CODED RESPONSES FROM PRE-STUDY

USER 1
Hi Primary researcher, I think all the points except number are valid and should remain on the list. I'm not sure what you mean by '2. Maximize a healthy environment.' Changing the wording might make what you are trying to get at clearer. hope this helps and hope all is well with you, User 1
USER 2
9. Minimize rent The reason is I think obvious :).... Cheers, -- User 2
USER 3
hey Primary researcher, Here are my thoughts.. something should be removed (which number) ---none something should be added (which number) ---the neighborhood is really important to me when looking for a place the wording needs to be changed so that it is clear immediately what is being talked about (which number) ---3 & 4, these are a little vague Let me know if you need more info :) User 3
USER 4
Minimise rent?
USER 5
One of the most critical criteria for an apartment (for me) is the utilities and appliances available: washer/dryer, dishwasher, modern bathroom, stove, oven, dual-basin sink in the kitchen, modern electrical power with abundant three-prong outlets, high-speed internet access, etc etc. Then there are the structural concerns: how old is the building? How much of a hazard does it present in an earthquake or fire? How well-insulated is it? (If you're paying your own utilities then cost to heat the space is a major issue. A modern, properly-insulated building will be much cheaper to heat) How clean is the building? Does the building have a flat roof? (A flat roof gets hotter in the summer, and can leak more easily. The famous "leaky condominiums" had flat roof panels that allowed water to run down the outside face of the walls. This let water seep into the walls, and the walls began to get moldy and rotten. This is expensive to fix, and poses a health hazard.) Structure also effects item #3: what is the temperature indoors? Do you have your own thermostat? How good is the lighting (both electrical and natural), direction/number of windows. (e.g. is there a south-facing window to get nice natural light?) Then of course there is PRICE. How much is the rent? Is it a month-to-month rental or a lease? Is the landlord/lady a reasonable person? (If they're a pain in the butt it can ruin an otherwise fantastic apartment.) Are utilities included? Parking? As with most practical problems there are a huge number of factors.

Hope this helps -User 5
USER 6
Minimize cost
USER 7
I would add the following, Minimize the cost Living near people with the same ethnic background Minimize distance to everday utilities (more important for ethnic minorities, such access to halal/kosher meat)
USER 8
Hi Primary researcher, In my opinion, you have covered all the factors that I can think of !! I am curious to know if there could be any more additions .. so it would be great if you could also compile pplz answers .. and send it around :) !! User 8
USER 9
Hi Primary researcher, It looks really good, I would recommend the following: Number 1: The impression I got was that this included commute to work time only. Either another entry needs to be added, or this one needs to be expanded, as there are other "work like" things that one might want to be minimize their distance to, such as grocery shopping, day care, etc. Number 6: In addition to wanting to be close to nature people might also have other priorities: close to nightlife, a particular neighborhood, restaurants, their family, etc. Finally, I'm not sure I understand the difference between 2 and 3&4. That's all I can think of, good luck. User 9
USER 10
You forgot cost! If you can smoke, have pets, babies, etc.
USER 11
I'd removed #5 since I don't think that there is a great variation in how energy efficient apartments are and it's hard to tell without having lived there for a while. For that factor it seems much more important what you put inside and you travel time and mode relative to conveniences. In the same vein, I'd added a new value which is proximity to non-work conveniences like supermarkets, bus routes, a good bar, and friends. This could perhaps be bundled in with proximity to nature. Cheers,

User 11

USER 12

Hi Primary researcher,

2. Might need to be clarified or removed. It seems very similar to either 5 or 6, but I don't understand why people would want their house to be "in a healthy environment" other then making the house more efficient (5), having an outdoors (4) or being close to nature (6).

+9. While certainly not important for some people, Maximizing proximity to schooling or day care might be important. You might factor this into commute time.

User 12

USER 13

Hi Primary researcher. Sounds like an interesting study.

Except for the commute time, I found your questions difficult to actually consider -- they're almost meta-questions. Or maybe it's that when thinking about places, I use a combination of analytical decision making as well as gut-based decision making.

I would actually consider neighbourhood vibrancy over commute time, generally, as long as the commute time is within reason. I lived in the West End which had a longer commute time than, say, South Granville, but I preferred:

the neighbourhood feeling (busy vs quiet, arty vs sterile)
the diversity of the neighbourhood (ethnic mix, age mix)
the easy availability and quality of day-to-day shopping needs
(e.g., green grocers, super market)

But none of those are captured directly by your questions. And I think there are also more negatives than just commute time. You haven't incorporate rent/cost, size.

Have you thought of asking several people why or why they wouldn't want to live in certain areas, ensuring to include some outlying areas. You could then ascertain the order of their preferences by varying different conditions: if Cloverdale is too far, would that change if there was a nearby SkyTrain station and a quick route into UBC? If rent is too high, would that change if they found a real bargain?

But I wonder if you'd only find out that different people have different preferences. After all, some people like living in the burbs, and others like living in Yaletown. Places I wouldn't want to live in! Actually, this sounds pretty complicated. Have you looked at the community planning literatures? I'm sure somebody must have done something about people's preferences.

Anyways, good luck with your study.

User 13.

USER 14

Hey Primary researcher,

While I'm not sure if these are things that can be maximized per se, they're still definitely things I anticipate considering when I'm looking for my next place. One, ability to keep pets. Not sure that can be maximized since it's just yes or no, but perhaps you can work that in somewhere. Two, desirable neighborhood. E.g. even a wonderful apartment in Surrey is still in Surrey. Regardless of how it affects my commute, there's no way I'm living in Surrey. It's simply too far from what I considering to be the interesting parts of the Vancouver area.

Aside from that, it looks good. I didn't see the actual dollar cost of rent, but I assume that's factored in somehow. Good luck and if you need any volunteers later, just let me know. Thanks!

-User 14

USER 15

you missed the cost of the apartment....you want to minimize cost.

USER 16

This is the list of fundamental objectives

>

1. Minimize commute time

>

I want commute time to be within a range, like under 1 hour. My current commute is 2 minutes, so when I look for a new place, I don't want another minimum commute.

>

1. Maximize a healthy environment
2. Maximize the comfort of the indoor living space

>

I don't understand what this means. The comfort would be based on too many things that are not necessarily a part of the apartment. Some want carpet, some want wood floors. TV, kitchen size, windows, furniture and many other things go into comfort level, and I don't see how you can express that all in some maximization formula.

>

1.
2. Maximize the comfort of the outdoor living space
3. Maximize energy efficiency
4. Maximize proximity to nature

>

Proximity to what type of nature? Some want the beach, some want the mountains, some want parks, or forests. Similarly, you should have a proximity to city things, like grocery stores, restaurants, bars, shopping, movie theaters, etc.

>

1. Maximize ability to entertain
2. Maximize safety

>

Also, you don't have standard apartment features. # of bedrooms, # of bathrooms, washer/dryer, kitchen, patio, balcony, yard, basement or not, apartment or house, roommates or not, furnished or unfurnished, length of lease, pets, which utilities are included, price range. Not everything can be expressed in terms of minimizing and maximizing.

USER 17

What about general cost? Or bedroom count? Or proximity to a major metropolitan area?

What does 1. mean? And how is 2. or 3. quantified when searching for a place to live? I can understand a basic space criteria, of "comfort" throws me.

User 17

USER 18

add Minimize cost
clarify if "healthy environment" means "my local environment is healthy for me" or "it contributes to the health of the planet, or at least minimizes damage"

add Minimize distance to {friends, family, bf/gf}
add Minimize distance to hobbies (e.g. skiing)
add Maximize access to {libraries, shopping, bus line, bank, restaurants}

USER 19

Hi Primary researcher,

Two I would add are:
-maximize (or minimize?) proximity to friends and family
-minimize costs (factoring in utilities, etc)

Good luck with the study...

Cheers,
User 19.

USER 20

Hi Primary researcher,

I would probably look for somewhere that has some recreational facility (such as pool or/and gym) as well. I think number 7 would refer to such things as party rooms, movie rooms or satellite dishes. Moreover I think it's better to change number 8 to 'maximize safety and security'. I think safety mostly refer to having for example a good
fire alarm system.
Cheers,
User 20

USER 21

hi, my name is jacob. and im looking for an apt.

the list isnt in order of importance in my opinion. but, it seems good. i guess if i were to say what is important to me, i would say:

minimize cost (factoring in salary), based on these factors:

location:

commute
safety
close to friends
closeness to public transportation (i.e. walking distance to subway)
city (some people want to be close to the city)
beach (some people want to be close to the beach)
stores (some people want groceries to be convenient)
some people want to be near certain things and i dont know how to phrase that

space:

size (spaciousness, whether everything jacob owns will fit inside)
interior quality (floors, walls, windows)
rooms (kitchen, bathroom, balcony, garage, laundry room)
view (I LOVE THIS ONE!!! obviously.. hehe)
higher floors usually have better views
yard, garden, whether it's in a nice complex, deck or patio
parking
building amenities (gym, daycare, pool, doorman)
exterior appearance (architecture, paint job)

ok, that's all i can think of. i know this is not in the format that you asked for, but maybe you can get some ideas from this if you want. but, the list you have covers most of these issues.

good luck with your survey! (im making ****** respond to you, so let me know if he doesnt)

User 21

USER 22

Hi,

One thing that is important for my wife and is the proximity to services (laundry, bank, clinic, mall, etc.), but this might fall under "Minimize commute time" already. I do think of "commute time" as being to school/work and back to home, and these others as something separate.

Hope this help,
User 22

USER 23

Hi,

I am somewhat confused about number 2. Do you mean maximize eco-friendliness, or a healthy living environment?

Best,
User 23

USER 24

Hi,
How about:
quiet environment
allow pets
non-smoking
sunlight (important in Vancouver)
room mates?
3 & 4 probably don't need the phrase: "the comfort of the".

User 24

USER 25

hi Primary researcher,

I think people might get confused about #2, some people might have different definitions of a healthy environment. Also, #7, do you mean entertain guests, or entertain yourself?
Just a suggestion, maybe you could provide an example with each objective.
For example:
Minimize commute time (You want a place that is near public transportation)
Maximize a healthy environment (You don't want to live near dirty, run-down streets with lots of homeless people)

etc...

User 25

USER 26

Primary researcher,

1. Minimize commute time

Seems self-explanatory to me.

2. Maximize a healthy environment

This seems to be a little too general. Does it include, say, the amount of natural light indoors (conflict with #3); the pollution--noise and otherwise--outside (conflict with #4)?

3. Maximize the comfort of the indoor living space
4. Maximize the comfort of the outdoor living space
5. Maximize energy efficiency
6. Maximize proximity to nature

Seems to be related (at least, to many people) with #7.

7. Maximize ability to entertain

This perhaps should be stated in terms similar to #6: proximity to cultural/shopping hot-spots, or some such.

8. Maximize safety

Thanks a lot (in advance)

Hope this is helpful. Good luck with your study.

USER 27

Hi Primary researcher,

Here are my two cents...

Add as #1: Minimize distance to the places where family (kids, parents) and friends live

Put "Maximize Safety" as #2. It's not a good idea to live in certain areas of the city.

There should be "Minimize commute time by car" AND "Minimize commute time by public transit", because commute time by car and by transit are often drastically different. Public transit is very helpful for teens, elderly people, and people working close to SkyTrain.

Commute time is less important than healthy environment. Noone wants to to have a highway exit or a SkyTrain station in their backyard.

I'm not sure whether "Comfort of the outdoor living space" includes insfrastructure like every-day-use municipal parks. I assume it does and the "Proximity to nature" refers to less often used wilderness like provincial park.

All in all, my list looks like
"Minimize distance to family/friends"
"Maximize safety"
"Maximize a healthy environment"
"Minimize commute time by public transit"
"Minimize commute time by car"
"Maximize the comfort of the indoor living space"
"Maximize the comfort of the outdoor living space"
"Maximize energy efficiency"
"Maximize proximity to nature"
"Maximize ability to entertain"

Hope that helps,

User 27

USER 28

Hi Primary researcher,
I think that you missed a major objective that is rent/price. That is probably most often the most important objective. I think that point 2 is vague. I am not sure how many people think explicitly about 5, they are more likely to deliberate over room layout (which I guess falls under 3) or different means of commuting than to think about 5. Also proximity to friends/family might be important to some people. I am not sure if any of your points addresses proximity to shopping places.
Cheers,

User 28

Appendix G. PROCEDURES AND INSTRUCTIONS FOR FINAL USER STUDY

Except for the first participant, click on Thank You for Your Time from the apartments interface to begin a new instance of the study when the user arrives

START

Have them sign the consent form

Read

The goal of this experiment is to explore a method for acquiring people's preferences. To start the study, I'd like to show you the sample interface. You will be learning how to interact with it.

INTERFACE TRAINING

Read

Suppose I want to buy a desk. I have preferences for my desk. For example, I would like an Has L - Shape or a desk with a keyboard-drawer. Often such preferences accomplish objectives that my desk will fulfill in my life, workplace, home etc.

Show sample interface

Read

For e.g., one of my objectives maybe to maximize aesthetics of my desk or improve ergonomics of my desk. When objectives are organized hierarchically it forms a value tree. On the left side is a value tree. I have been provided with a value tree to begin with. I can personalize this value tree to reflect my preferences.
My objectives can be fulfilled by things in the real world, i.e. things such as 'made of wood', 'wide keyboard drawer' etc. We call these elements in the real world, attributes. Most often attributes belong to a class that is a set of objects, known as entities such as 'desk', 'drawer' etc. These entities and attributes often have a hierarchical representation in the real world – we call this domain ontology. On the right side is a domain ontology for desks.

Do you have any questions?

PART 1: VALUE TREE AND DOMAIN ONTOLOGY

Read
Let us begin part 1. This interface allows me to represent my preferences in terms of objectives and attributes that accomplish those objectives. What you see in front of you is Part 1 of the training interface – this allows me to express my value tree. Suppose I would like to maximize my workspace, maximize the ergonomics of desk, minimize costs and maximize the aesthetics of my desk. Let's add the first objective to my value tree.

How to add an objective to the value tree?
To add an objective, I click on the text box or the label "Add an objective". Type my objective. MAXIMIZE ERGONOMICS OF DESK Click "Add" to add the objective to the value tree below.

How about you do it now?

Add the objective: MAXIMIZE WORK SPACE

How to modify an objective in the value tree?
I have decided I want to improve the ergonomics of my workspace. To modify an objective, click on the pencil [✐] icon next to the objective, in this case the MAXIMIZE ERGONOMICS OF DESK objective is what I wish to change. An inline-editing textbox will appear where the objective was. I can modify the objective to be IMPROVE ERGONOMICS OF DESK Once I am done modifying the objective, I can click 'ok' to save the objective or 'cancel' to not make any modifications. I'll click 'ok' and save the changes.

How about you do it now?

Modify the objective: MAXIMIZE WORK SPACE to be MAXIMIZE STORAGE SPACE

How to remove an objective in the value tree?
To remove an objective from the value tree, click on the red cross [⊗] icon next to the objective you wish to remove. Suppose I decided that space is no

longer a constraint in my decision making and would like to remove the objective:
MAXIMIZE STORGE SPACE
A message will ask me to confirm whether I choose to remove the objective.
Clicking 'OK' will remove the objective from the value tree.
Clicking 'cancel' will not remove the objective from the value tree.
I'll click okay to remove the objective.

How about you do it now?

Remove the objective: MINIMIZE COSTS

Suppose the objective: IMPROVE ERGONOMICS OF DESK may be fulfilled by the following attributes: Has Wide Keyboard Drawer, Has L - Shape and Made of wood. Let's add these attributes.

How to add an attribute to fulfill an objective?

Drag the attribute from the ontology tree. I'll choose Has Wide Keyboard Drawer from the Ontology on the right hand side. Drop the attribute under the objective: IMPROVE ERGONOMICS OF DESK.
When I hover with a selected attribute over an objective, a highlighted box will appear around the objective.
If the attribute is dropped in this box, the attribute will be added under that objective.
If the attribute is not dropped in such a box, the attribute will not be added to any objective.

How about you do it now?

Add the attribute Has L - Shape to the objective: IMPROVE ERGONOMICS OF DESK.
Add the attribute Made of wood to the objective: IMPROVE ERGONOMICS OF DESK.

How to move an attribute from one objective to another?

Say I decided that if the desk is Made of wood it would fulfill the aesthetic objective of My desk rather than the ergonomic so I would like to move the attribute from one objective to another.
I drag the attribute label Made of wood from the objective: IMPROVE ERGONOMICS OF DESK to the objective: MAXIMIZE AESTHETICS OF DESK.
If the attribute is not dragged to any other objective's highlighted box, the attribute will return to the original objective.

How about you do it now?

Drag the attribute Has L - Shape from the objective: IMPROVE ERGONOMICS OF DESK to the objective: MAXIMIZE AESTHETICS OF DESK.

How can I observe the relative positions of the attributes, under an objective, in the Ontology?

Often an ontology can be quite large and difficult to navigate. To see where the relative positions of the attributes, under an objective, is in the domain ontology we have added this feature.
Click the objective label: MAXIMIZE AESTHETICS OF DESK to see where the relative positions of the attribute Has L – Shape, Made of wood is in the ontology.
This will highlight the attribute in the ontology tree.

How can I reset the highlighting in the Ontology?

Clicking on the objective label that is already highlighted will remove the highlighting. Go ahead and click on the objective: IMPROVE ERGONOMICS OF DESK to reset the highlighting for the objective MAXIMIZE AESTHETICS OF DESK and highlight the attributes of the new objective clicked.

How about you do it now?

Click on MAXIMIZE AESTHETICS OF DESK to highlight the attributes in the ontology.

Suppose to IMPROVE ERGONOMICS OF DESK, the attribute Has L - Shape is necessary. Since the attribute, Has L - Shape is already under one objective it is easy to find its position in the ontology as it is highlighted.
Add the attribute Has L - Shape to the objective IMPROVE ERGONOMICS OF DESK.

However, to MAXIMIZE AESTHETICS OF DESK I don't want to get a desk made of plastic.
Add the attribute made of plastic to the objective MAXIMIZE AESTHETICS OF DESK.

When I double click on this attribute it changes color from green to red and now says made of plastic is false which means that not having this attribute would specifically fulfill the corresponding objective MAXIMIZE AESTHETICS OF DESK.

How to remove an attribute from an objective?

Say I then decided the aesthetics of my desk does not really depend on whether it is made of wood. Therefore, I would like to remove the attribute Made of wood from the objective: MAXIMIZE AESTHETICS OF DESK.
Click the red cross [⊗] icon next to the attribute label of Made of wood under the objective: MAXIMIZE AESTHETICS OF DESK.
A message will ask me to confirm whether I choose to remove the attribute from that objective.
Clicking 'OK' will remove the attribute from the value tree.
Clicking 'cancel' will not remove the attribute from the value tree.
I'll click okay to remove the attribute from the objective.

How about you do it now?

Remove the attribute Has Wide Keyboard Drawer from the objective: MAXIMIZE AESTHETICS OF DESK.

Do you have any questions?

PART 2: VALIDATION

Read

To validate the value tree created in part 1, we will ask some questions in part 2, which will allow us to understand the structure of my true preferences.
Before we move onto part 2 – here is some training to get you used to the structure of these questions.

Hand the training document and read

The questions are in the comparison gamble format, which is a paired comparison. You must choose between two alternatives or state you are indifferent between the two alternatives.

When the user has handed the training, read

To validate the value tree created in part 1, we will ask some questions in part 2, which will allow us to understand the structure of my true preferences.
I will do the first one and you can do the next one.

Do you have any questions?

Click on Thank You for Your Time from the desk interface to reset the training module when the new user arrives

SCENARIO

Let us begin the study
Read

Suppose, you are relocating to Vancouver, British Columbia Canada. You have preferences when it comes to renting an apartment. You are given a fixed budget of $1000 for rent. For the purposes of the study you do not need to consider minimizing your costs as an objective.

Start the program
Read

The following interface is provided for you to express your preferences in terms of your objectives and the attributes in the apartment's domain that fulfill these objectives. On the left hand side is the default value tree that specifies a set of objectives. You do not have to stick with this set of objectives. You are free to modify, add or remove any of these objectives as you see fit.

Point out part 1 (Mapping)

This is Part 1 of the interface.
During this part 1, if you have any questions please do not hesitate to ask.
Once you have completed this section, please proceed onto the next part – Part 2, the Validation.

CONCLUDING STUDY

Thank them, Pay, and Have them sign receipt

Appendix I. USER INFORMATION

THE UNIVERSITY OF BRITISH COLUMBIA

Department of Computer Science
2366 Main Mall
Vancouver, B.C., V6T 1Z4

Value-Focused Preference Model Construction Given A Domain Ontology

Age: ☐ 19 or below ☐ 20 – 30 ☐ 31 – 40 ☐ 41 – 50 ☐ 51 – 60 ☐ 61 and above

Gender : ☐ Male ☐ Female

Occupation: ____

Have you ever rented an apartment? ☐ Yes ☐ No

Have you ever purchased a desk? ☐ Yes ☐ No

Appendix J. VALIDATION QUESTION TRAINING

Part 1

You will be asked to determine probabilities that make you indifferent between pairs of outcomes. Outcomes can be certain or uncertain. In addition, they can be gains or losses.

THERE IS NO RIGHT OR WRONG ANSWER. It all depends on your preference and your attitude towards risk.

Here is a sample question.

Please determine the value of *p* **that would make you indifferent** between **OPTION A** and **OPTION B**.
OPTION A: to enter a lottery in which you have a *p* chance of winning **a free coffee and a muffin** and a (1-p) chance of winning **nothing**.
OPTION B: to win **a free coffee.**
A simple table like this can help your assessment by trying different values of *p* until you find one that makes you indifferent between OPTION A and OPTION B.
Please write down in the table all the values of p that you consider and your corresponding preferred outcome.
For example, I prefer OPTION B, if the value of p is 0, because I would rather get a free coffee than nothing. However, I prefer OPTION A, if the value of p is 1, because I would rather get a free coffee and a free muffin as there is no risk.

p	*1-p*	*I prefer ...*
0	1	**OPTION B**
1	0	**OPTION A**
0.5	0.5	

When you are done, write down the value of *p*:____
To help you visualize your choices, the pairs of outcomes will be shown in a grid. The two outcomes are separated onto the left and right hand side. If an outcome is a lottery, a branching specifies probabilities and rewards of each lottery prize. For example, the previous example would be visualized as:

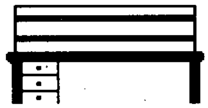
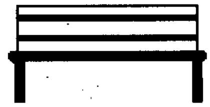
OPTION A		**OPTION B**
p	Free coffee and muffin ☺	Free coffee ☺
1-p	Nothing ☹	

Part 2

Let us focus on the desk domain. You are searching for a new desk. You are trying to come up with the best balance for your storage needs. Assume that the desk has the exact same dimensions for all alternatives.

For which value of p would you be indifferent between **OPTION A** and **OPTION B**? Please write down in the table all the values of p that you consider and your corresponding preferred outcome. Again, remember, in this study THERE ARE NO RIGHT OR WRONG ANSWERS, it all depends on your preferences and your attitude towards risk.

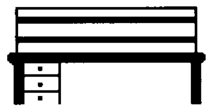
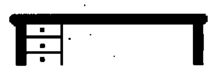
p	*1-p*	*I prefer ...*
0.5	0.5	______

<table>
<tr><th colspan="2">OPTION A</th><th>OPTION B</th></tr>
<tr><td>p</td><td>Desk with 3 drawers and 3 shelves ☺
</td><td rowspan="2">Desk with 3 shelves ☺
</td></tr>
<tr><td>1-p</td><td>Desk with no drawers or shelves ☹
</td></tr>
</table>

p that makes you indifferent ______

For which value of p would you be indifferent between **OPTION A** and **OPTION B**? Again, please write down in the table all the values of p that you consider and your corresponding preferred outcome.

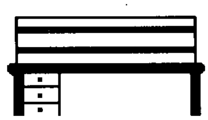
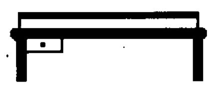
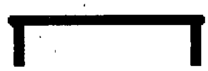
p	*1-p*	*I prefer ...*
0.5	0.5	______

<table>
<tr><th colspan="2">OPTION A</th><th>OPTION B</th></tr>
<tr><td>p</td><td>Desk with 3 drawers and 3 shelves ☺
</td><td rowspan="2">Desk with 3 drawers ☺
</td></tr>
<tr><td>1-p</td><td>Desk with no drawers or shelves ☹
</td></tr>
</table>

p that makes you indifferent ______

For which value of p would you be indifferent between **OPTION A** and **OPTION B**? Again, please write down in the table all the values of p that you consider and your corresponding preferred outcome.

p	*1-p*	*I prefer ...*
0.5	0.5	______

<table>
<tr><th colspan="2">OPTION A</th><th>OPTION B</th></tr>
<tr><td>p</td><td>Desk with 3 drawers and 3 shelves ☺
</td><td rowspan="2">Desk with 1 drawer and 1 shelf ☺
</td></tr>
<tr><td>1-p</td><td>Desk with no drawers or shelves ☹
</td></tr>
</table>

p that makes you indifferent ______

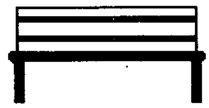
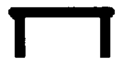
Part 3

Let us focus on the desk domain. You are searching for a new desk. You are trying to come up with the best balance for your storage needs. Assume that a *large desk* is best suitable for your needs, a *medium desk* will be a tradeoff and a *small desk* is going to be difficult to work with.

The grid on the left hand side now has an extra column for **OPTION B**. The percentages for **OPTION B,** however, are always the same, 50%. Consider two aspects of a desk. One is the ***number of shelves*** and the other is the ***size of the desk***.

For which value of *p* would you be indifferent between **OPTION A** and **OPTION B**? Please write down in the table all the values of *p* that you consider and your corresponding preferred outcome. Again, remember, in this study THERE ARE NO RIGHT OR WRONG ANSWERS, it all depends on your preferences and your attitude towards risk.

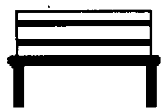
p	*1-p*	*I prefer ...*
0.5	0.5	

OPTION A		OPTION B	
p	Large desk with 3 shelves ☺ 	50%	Small desk with 3 shelves 
1-p	Small desk with no shelves ☹ 	50%	Large desk with no shelves 

p that makes you indifferent ______________

For which value of *p* would you be indifferent between **OPTION A** and **OPTION B**? Please write down in the table all the values of *p* that you consider and your corresponding preferred outcome.

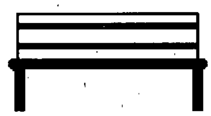
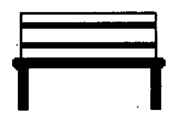
p	*1-p*	*I prefer ...*
0.5	0.5	

OPTION A		OPTION B	
p	Medium desk with 3 shelves ☺ 	50%	Small desk with 3 shelves 
1-p	Small desk with no shelves ☹ 	50%	Medium desk with no shelves 

p that makes you indifferent ______________

For which value of *p* would you be indifferent between **OPTION A** and **OPTION B**? Please write down in the table all the values of *p* that you consider and your corresponding preferred outcome.

p	*1-p*	*I prefer ...*
0.5	0.5	

OPTION A		OPTION B	
p	Large desk with 3 shelves ☺ 	50%	Medium desk with 3 shelves ☺ 
1-p	Medium desk with no shelves 	50%	Large desk with no shelves 

p that makes you indifferent ______________

Appendix K. POST-STUDY QUESTIONNAIRE

THE UNIVERSITY OF BRITISH COLUMBIA

Department of Computer Science
2366 Main Mall
Vancouver, B.C., V6T 1Z4

Value-focused preference model construction given a domain ontology

Do you have any suggestions/comments/questions about the Default Value Tree provided?

Do you have any suggestions/comments/questions about the Ontology provided?

Do you have any suggestions/comments/questions about the interface?

Other suggestions/comments/questions:

Appendix L. SAMPLE RECEIPT

University of British Columbia
Department of Computer Science

I, ____

have received the sum of $________, as payment for my participation in the "**Value-focused preference model construction given a domain ontology**" user study.

Signature ______________________________ Date__________________

Appendix M. UBC RESEARCH ETHICS BOARD CERTIFICATES OF APPROVAL

The University of British Columbia
Office of Research Services
Behavioural Research Ethics Board
Suite 102, 6190 Agronomy Road,
Vancouver, B.C. V6T 1Z3

CERTIFICATE OF APPROVAL- MINIMAL RISK RENEWAL

PRINCIPAL INVESTIGATOR:	DEPARTMENT:	UBC BREB NUMBER:
Giuseppe Carenini	UBC/Science/Computer Science	H05-80581

INSTITUTION(S) WHERE RESEARCH WILL BE CARRIED OUT:

Institution	Site
UBC	Vancouver (excludes UBC Hospital)

Other locations where the research will be conducted:
N/A

CO-INVESTIGATOR(S):
Blair Tennessy
Jeanette L. Bautista

SPONSORING AGENCIES:
Natural Sciences and Engineering Research Council of Canada (NSERC) - "Multimedia Generation and Interactive Visualization of Preferential Choice"

PROJECT TITLE:
Multimedia Generation and Interactive Visualization of Preferential Choice

EXPIRY DATE OF THIS APPROVAL: September 10, 2008

APPROVAL DATE: September 10, 2007

The Annual Renewal for Study have been reviewed and the procedures were found to be acceptable on ethical grounds for research involving human subjects.

Approval is issued on behalf of the Behavioural Research Ethics Board

Appendix N. APARTMENT ONTOLOGY IN OWL

```
<?xml version="1.0"?>
<!DOCTYPE owl [
<!ENTITY MILO
"http://www.cs.ubc.ca/~albaab01/thesis/MILO.owl#">
<!ENTITY dc "http://purl.org/dc/elements/1.1/">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
<!ENTITY SUMO
"http://www.cs.ubc.ca/~albaab01/thesis/SUMO.owl#">
<!ENTITY APRT
"http://www.cs.ubc.ca/~albaab01/thesis/APRT.owl#">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
]>
<rdf:RDF

xmlns:MILO="http://www.cs.ubc.ca/~albaab01/thesis/MILO.owl#
"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:SUMO="http://www.cs.ubc.ca/~albaab01/thesis/SUMO.owl#
"

xmlns:APRT="http://www.cs.ubc.ca/~albaab01/thesis/APRT.owl#
"
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:owl="http://www.w3.org/2002/07/owl#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://www.cs.ubc.ca/~albaab01/thesis/APRT.owl"
```

```
>
<owl:Ontology
rdf:about="http://www.cs.ubc.ca/~albaab01/thesis/APRT.owl">
<rdfs:label>APRT</rdfs:label>
<owl:imports
rdf:resource="http://www.cs.ubc.ca/~albaab01/thesis/MILO.ow
l"/>
<owl:imports
rdf:resource="http://www.cs.ubc.ca/~albaab01/thesis/SUMO.ow
l"/>
</owl:Ontology>
<owl:Class rdf:about="#Appliance">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;StationaryArtifact">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Balcony">
</owl:Class>
<owl:Class rdf:about="#BusTerminal">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#TransitTerminal">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Carpet">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#FloorType">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Den">
```

```
</owl:Class>
<owl:Class rdf:about="#Facility">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;StationaryArtifact">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#FerryTerminal">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#TransitTerminal">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#FloorType">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;Artifact">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Foyer">
</owl:Class>
<owl:Class rdf:about="#Good-View">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ViewAttribute">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Gymnasium">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&MILO;SportsFacility">
    </owl:Class>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:about="#Hallway">
</owl:Class>
<owl:Class rdf:about="#Hardwood">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#FloorType">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Linoleum">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#FloorType">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#LivingRoom">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Mountain-view">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Good-View">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#No-View">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#ViewAttribute">
    </owl:Class>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:about="#Ocean-view">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Good-View">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Office">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#TransitTerminal">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Tree-view">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Good-View">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#ViewAttribute">
  <rdfs:subClassOf>
    <owl:Class rdf:about="&SUMO;RelationalAttribute">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Vinyl">
```

```
  <rdfs:subClassOf>
    <owl:Class rdf:about="#FloorType">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;AerobicExerciseDevice">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;ApartmentBuilding">
</owl:Class>
<owl:Class rdf:about="&MILO;ApartmentUnit">
</owl:Class>
<owl:Class rdf:about="&MILO;Chair">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Chimney">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Door">
</owl:Class>
<owl:Class rdf:about="&MILO;EducationalFacility">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
```

```
      </owl:Class>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="&MILO;FanDevice">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Appliance">
      </owl:Class>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="&MILO;Fireplace">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Appliance">
      </owl:Class>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="&MILO;Floor">
  </owl:Class>
  <owl:Class rdf:about="&MILO;Mailbox">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Appliance">
      </owl:Class>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="&MILO;Monument">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Facility">
      </owl:Class>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="&MILO;Oven">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Appliance">
```

```
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Paint">
</owl:Class>
<owl:Class rdf:about="&MILO;ParkingLot">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;PerformanceStage">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;SecurityAlarm">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;SportsFacility">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;SportsGround">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
```

```
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Stove">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;SwimmingPool">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;TelephoneLine">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Toilet">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Wallpaper">
</owl:Class>
<owl:Class rdf:about="&MILO;WashingDevice">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Appliance">
```

```
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&MILO;Window">
</owl:Class>
<owl:Class rdf:about="&MILO;WindowCovering">
</owl:Class>
<owl:Class rdf:about="&SUMO;Artifact">
</owl:Class>
<owl:Class rdf:about="&SUMO;DirectionalAttribute">
</owl:Class>
<owl:Class rdf:about="&SUMO;EducationalOrganization">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&SUMO;MercantileOrganization">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Facility">
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="&SUMO;RelationalAttribute">
</owl:Class>
<owl:Class rdf:about="&SUMO;Room">
</owl:Class>
<owl:Class rdf:about="&SUMO;StationaryArtifact">
</owl:Class>
<owl:Class rdf:about="&owl;Thing">
</owl:Class>
<owl:ObjectProperty rdf:about="#hasApartmentUnit">
```

```
  <rdfs:domain>
    <owl:Class rdf:about="&MILO;ApartmentBuilding">
    </owl:Class>
  </rdfs:domain>
  <rdfs:range>
    <owl:Class rdf:about="&MILO;ApartmentUnit">
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasRoom">
  <rdfs:domain>
    <owl:Class rdf:about="&MILO;ApartmentUnit">
    </owl:Class>
  </rdfs:domain>
  <rdfs:range>
    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasWindow">
  <rdfs:domain>
    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
  </rdfs:domain>
  <rdfs:range>
    <owl:Class rdf:about="&MILO;Window">
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="#closeToAirport">
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#closeToArtSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBank">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBaseballField">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBasketballCourt">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBeach">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBusStop">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBusTerminal">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToBusinessSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToCollege">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToDaySchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToDrugStore">
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#closeToEducationalOrganization">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToFacility">
  <rdfs:domain>    <owl:Class
rdf:about="&MILO;ApartmentBuilding">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToFerryTerminal">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToGraduateSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToGroceryStore">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToGymnasium">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToHighSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToJuniorCollege">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToLaundromat">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToLawSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToLibrary">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToLiveTheatre">
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#closeToMajorRoadIntersection">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToMedicalSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToMovieTheatre">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToMuseum">
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToPark">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToParkingLot">
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#closeToPostSecondarySchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToPrivateSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToPublicLibrary">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToPublicSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToPubs">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToRailwayTerminal">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToRestaurants">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToRetailStore">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToSecondarySchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToShoppingMall">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToSoccerField">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToSportsFacility">
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#closeToSportsGround">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToSwimmingPool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToTheate">
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#closeToTouristAttraction">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToTransitTerminal">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToUniversity">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToVocationalSchool">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#closeToWholesaleStore">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#facesEast">
  <rdfs:domain>    <owl:Class rdf:about="&MILO;Window">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#facesNorth">
  <rdfs:domain>    <owl:Class rdf:about="&MILO;Window">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#facesSouth">
```

```
  <rdfs:domain>    <owl:Class rdf:about="&MILO;Window">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#facesWest">
  <rdfs:domain>    <owl:Class rdf:about="&MILO;Window">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty
rdf:about="#hasAerobicExerciseDevice">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasAirConditioning">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasAppliance">
  <rdfs:domain>    <owl:Class
rdf:about="&MILO;ApartmentUnit">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasAreaMeasure">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasBalcony">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasBathroom">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasBathtub">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasBedroom">
  <rdfs:subPropertyOf rdf:resource="#hasRoom" />
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasCarpet">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasChair">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasChimney">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasCloset">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDen">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDiningRoom">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDishwasher">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDoor">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDressingRoom">
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasDryer">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFacility">
  <rdfs:domain>    <owl:Class
rdf:about="&MILO;ApartmentBuilding">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFan">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFireplace">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFlooring">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFoyer">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasFridge">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasGrounds">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasGymnasium">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasHallway">
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasHardwood">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasKitchen">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLargeArea">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLaundryFacility">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLibrary">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLinoleum">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLivingRoom">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasMailbox">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasMediumArea">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasMountainView">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasOceanView">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasOffice">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasOilPaint">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasOven">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasPaint">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
```

```
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasPainting">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasPark">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasParkingGarage">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasParkingLot">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSecurityAlarm">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasShower">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSink">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSkylight">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSmallArea">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSportsFacility">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasStove">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSwimmingPool">
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasTelephoneLine">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasToilet">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasTreeView">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasView">
  <rdfs:domain>    <owl:Class
rdf:about="&MILO;ApartmentUnit">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasVinyl">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasWallpaper">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasWashBasin">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasWasher">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasWatercolorPaint">
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasWindowCovering">
  <rdfs:domain>    <owl:Class rdf:about="&SUMO;Room">
    </owl:Class>
```

```
</rdfs:domain>
  <rdfs:range>    <rdfs:Datatype rdf:about="&xsd;boolean"/>
</rdfs:range>
</owl:DatatypeProperty>


<owl:DatatypeProperty rdf:about="#closeToVocationalSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToLawSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToGraduateSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToSportsFacility">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBusTerminal">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToBusStop">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasCarpet">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFlooring">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToGymnasium">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSportsFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToPublicSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToTheate">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasMountainView">
  <rdfs:subPropertyOf>
```

```
  <owl:DatatypeProperty rdf:about="#hasView">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToCollege">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToUniversity">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToPubs">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasSmallArea">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAreaMeasure">
  </owl:DatatypeProperty>
```

```
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToPublicLibrary">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToLibrary">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasPark">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasAirConditioning">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBeach">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasMediumArea">
```

```
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAreaMeasure">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasOceanView">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasView">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasGrounds">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToGraduateSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToUniversity">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasWatercolorPaint">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasPaint">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasLaundryFacility">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasLinoleum">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFlooring">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasMailbox">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasDishwasher">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToMovieTheatre">
  <rdfs:subPropertyOf>
```

```
    <owl:DatatypeProperty rdf:about="#closeToTheate">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToDaySchool">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToSchool">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToGroceryStore">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToRetailStore">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToPark">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToFacility">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToRailwayTerminal">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty
  rdf:about="#closeToTransitTerminal">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasLibrary">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToMedicalSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToGraduateSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToRestaurants">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasToilet">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasOven">
```

```
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasFireplace">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToDrugStore">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToRetailStore">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasFan">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasTreeView">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasView">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
```

```
</owl:DatatypeProperty>

<owl:DatatypeProperty
rdf:about="#closeToTouristAttraction">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty
rdf:about="#closeToMajorRoadIntersection">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasParkingGarage">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToMuseum">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToEducationalOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#closeToSwimmingPool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSportsFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToWholesaleStore">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBaseballField">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSportsGround">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasChimney">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasSportsFacility">
  <rdfs:subPropertyOf>
```

```
    <owl:DatatypeProperty rdf:about="#hasFacility">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToLiveTheatre">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToTheate">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToPrivateSchool">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToSchool">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty rdf:about="#closeToUniversity">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty
  rdf:about="#closeToPostSecondarySchool">
    </owl:DatatypeProperty>
    </rdfs:subPropertyOf>
  </owl:DatatypeProperty>

  <owl:DatatypeProperty
  rdf:about="#closeToPostSecondarySchool">
    <rdfs:subPropertyOf>
    <owl:DatatypeProperty rdf:about="#closeToSchool">
    </owl:DatatypeProperty>
```

```
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBusStop">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToTransitTerminal">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty
rdf:about="#closeToEducationalOrganization">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasStove">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasGymnasium">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasSportsFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#closeToSoccerField">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSportsGround">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasChair">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToJuniorCollege">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToPostSecondarySchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToHighSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasParkingLot">
  <rdfs:subPropertyOf>
```

```
  <owl:DatatypeProperty rdf:about="#hasFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToShoppingMall">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBusinessSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToGraduateSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToEducationalOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasShower">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
```

```
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasLargeArea">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAreaMeasure">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasWasher">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasFridge">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToAirport">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToTransitTerminal">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasDryer">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty
rdf:about="#hasAerobicExerciseDevice">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToParkingLot">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasVinyl">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFlooring">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToSecondarySchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSchool">
```

```
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToRetailStore">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToMercentileOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasHardwood">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasFlooring">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasSecurityAlarm">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasSink">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#hasTelephoneLine">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasAppliance">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToLaundromat">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToSportsGround">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasOilPaint">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasPaint">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBank">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
```

```
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#hasSwimmingPool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#hasSportsFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToArtSchool">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSchool">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToLibrary">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToEducationalOrganization">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToBasketballCourt">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToSportsGround">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="#closeToFerryTerminal">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty
rdf:about="#closeToTransitTerminal">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="#closeToTransitTerminal">
  <rdfs:subPropertyOf>
  <owl:DatatypeProperty rdf:about="#closeToFacility">
  </owl:DatatypeProperty>
  </rdfs:subPropertyOf>
</owl:DatatypeProperty>
</rdf:RDF>
```

When an element is defined in an ontology that has been imported it is referred to by appending the symbol defined in the header of the OWL file for the ontology to the name of the entity. For instance, the **APARTMENT-UNIT** entity in MILO is referred to as `rdf:about="&MILO;ApartmentUnit"`. The imported SUMO and MILO ontologies can be found at http://ontology.teknowledge.com/#download