An Application Intersection Marketing Ontology

Xuan Zhou[1], James Geller[1], Yehoshua Perl[1], and Michael Halper[2]

[1] CS Department, New Jersey Institute of Technology, Newark, NJ 07102*
{xxz1279, geller, perl}@oak.njit.edu
[2] Mathematics & Computer Science Department, Kean University, Union, NJ 07083
mhalper@kean.edu

Abstract. We consider the design of an ontology for marketing knowledge. Such an ontology contains two hierarchies, a customer hierarchy and a product hierarchy. The product hierarchy representation is straightforward, as in general each level consists of products that are more specific than the products on the previous level. However, the customer hierarchy is problematic, since it involves many independent dimensions such as age, gender, income, *etc.* A straightforward ordering of the different dimensions to create a tree hierarchy is ineffective. We present an innovative design for the customer hierarchy based on introducing intersections of options for various dimensions *on demand*. We call such an ontology an intersection ontology. The advantages of such a design are explored and evaluated using our Web marketing project.

1 Introduction

1.1 Motivation

The result of market research is *marketing knowledge* that is used as input for target marketing activities. However, marketing knowledge is usually complex, consisting of many detailed facts, which by themselves do not give any clear picture and in combination are often overwhelming. What is desirable is an organization of marketing knowledge in an ontology that allows for the explicit representation of interesting abstractions and generalizations.

Ontologies have become important resources in many application domains. However, in marketing, ontologies have been close to non-existent. In this paper, we develop a kind of ontology, called an intersection ontology, for a marketing application and explore its advantages.

1.2 What Are Ontologies?

We will start with a fairly non-technical summary of what ontologies are and what they are useful for. Ontology is known as the branch of philosophy concerned with the study of the nature of being. However, when computer scientists

* This research was funded in part by the New Jersey Commission for Science and Technology through the New Jersey Center for Software Engineering.

O. Goldreich et al. (Eds.): Shimon Even Festschrift, LNCS 3895, pp. 143–163, 2006.

are referring to "an ontology" they mean a computer implementation of human-like knowledge.

Ontologies are descendants of the semantic networks in Artificial Intelligence. Quillian's first semantic network in 1968 was a computer implementation of a dictionary [1]. Terms in dictionaries refer to other terms, and Quillian implemented these references by pointers. However, as one term could have different meanings, a distinction is made between terms and concepts. Concepts are the fundamental building blocks of all semantic networks and ontologies.

A concept is a basic unit of knowledge, and, as opposed to a term, a concept is unambiguous. Quillian used only a small number of kinds of links which have been extensively studied and greatly refined since then. The most fundamental of these links, describes a generalization/specialization relationship between two concepts. This relationship satisfies transitivity. It has been variously called IS-A, sub-concept, subclass, a-kind-of, *etc.* It allows property inheritance, as follows.

Humans have additional "local" information about concepts. For example, solid objects have color, size, *etc.* We call this kind of local information "attributes", "properties" or "slots". If a general concept has an attribute (vehicles have a weight), then a specific sub-concept will have the same property (cars have a weight). One can imagine that inheritance is the propagation of a property from the general concept to the more specific concept against the direction of the IS-A link. Besides the IS-A links, ontologies contain other links, e.g., likes, owns, connected-to, *etc.* Most of these additional links have no "built-in behavior". These links are variously called associative relationships, roles, semantic relationships, and are labeled by their name. Relationships are inherited down along IS-A links.

Because a concept cannot be more general than itself, and because of the transitivity of the IS-A links, there cannot be any cycles of IS-A links in a semantic network. Furthermore, it is practical to have one concept (often called THING) that is a generalization of every concept in an ontology. Thus, the concepts and IS-A links in an ontology form a hierarchy with a root. In other words, the hierarchy of an ontology is a rooted Directed Acyclic Graph (DAG), where the nodes represent the concepts and the links represent IS-A relationships. Furthermore, the concepts and the IS-A links together would form a weakly connected component.

The definition of an ontology as a graph results in a natural diagram representation for ontologies. Figure 1 shows an example of an ontology. This example is adapted from [2] by eliminating other relationships such as part-of. In this and later figures, every box stands for a concept. Bold arrows (typically pointing upwards) stand for IS-A relationships. Thin arrows stand for other relationships. The IS-A relationships in this example form a tree. Later we will see examples using DAGs. Family terms, such as *child*, *ancestor* and *descendant* are used. A number of other extensions exist for ontologies, e.g. rules or axioms. However, these are not used in our model of an intersection ontology and will be omitted.

Thus, we present the definition of an ontology as follows: An ontology is a directed graph of nodes, which represent concepts, and edges, which represent

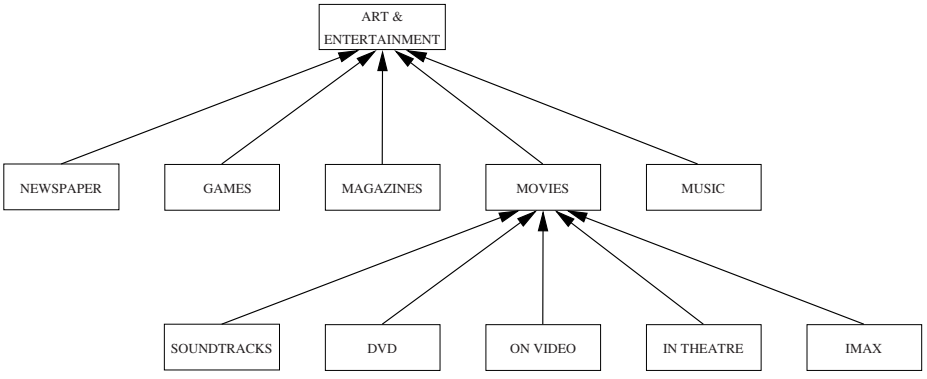


Fig. 1. A Partial Interest Hierarchy (Tree)

IS-A and semantic relationships between pairs of nodes. Concepts are labeled by unique terms. Concepts have additional (name, value) pairs, called attributes, where the attribute name needs to be unique for each concept. The set of all concepts together with the set of all IS-A links form a rooted, connected, Directed Acyclic Subgraph of the ontology. This subgraph is called the taxonomy of the ontology. Both attributes and semantic relationships may be inherited downwards, against the direction of the IS-A links, from more general concepts to more specific concepts.

Modern ontologies are attributed to Thomas Gruber [3] who built on a rich history which we briefly reviewed in [4]. Ontology building deals with modeling the world with shareable knowledge structures. With the emergence of the Semantic Web, the development of ontologies and ontology integration have become very important [5–8]. The Semantic Web is a vision, for a next generation Web, of Tim Berners-Lee, the inventor of the original Web, and colleagues. This vision is described in a figure called the "layer cake" of the Semantic Web [5]. This figure consists of nine functional layers of increasing technical complexity and abstraction. Each layer supports all the layers above it. Ontologies are flush in the middle of the layer cake. All the layers below Ontology, such as XML and RDF Schema are well developed. All the layers above ontologies, such as Rules and Proofs are well established within Artificial Intelligence (AI), but do not exist in widely applicable form outside of AI.

Ontologies will be used in the Semantic Web as follows. The current Web has shown that string matching by itself is often not sufficient for finding specific concepts. Rather, special programs are needed that search the Web for the concepts specified by a user. Such programs, which are activated once and traverse the Web without further supervision, are called agent programs.

Successful agent programs will search for concepts as opposed to words. Due to the well known homonym and synonym problems, it is difficult to select between different concepts expressed by the same word (e.g., Jaguar the animal, or Jaguar the car). However, having additional information about a concept,

such as which concepts are related to it, makes it easier to solve this matching problem. For example, if that Jaguar that IS-A car is desired, then the agent knows which of the meanings to look for.

Ontologies provide a repository of this kind of relationship information. To make the creation of the Semantic Web easier, Web page authors will derive the terms of their pages from existing ontologies, or develop new ontologies for the Semantic Web.

Many technical problems remain for ontology developers, e.g. scalability. Yet, it is obvious that the Semantic Web will never become a reality if ontologies cannot be developed to the point of functionality, availability and reliability comparable to the existing components of the Web.

Some ontologies are used to represent the general world or word knowledge. Other ontologies have been used in a number of specialized areas. An overview of ontologies and their usages and properties can be found in [9]. For a comprehensive review of established ontologies see [10]. Two special issues on ontologies are [11, 8].

1.3 Ontologies in the Context of Web Marketing

Our work on marketing ontologies is part of a larger project that deals with the extraction of marketing knowledge from the World-Wide Web [12]. We have created a large database of customers. We extracted information from the home pages of individual Web users. Our database contains demographic information and interests of each customer.

We would prefer information about products that each of these customers has bought. However, this information is not publicly accessible on the Web. On the other hand, there are many very low-level interests with corresponding products. For example, the Yahoo interest hierarchy contained over 31,000 interests when we analyzed it. Many of these interests are as specific as the names of actresses or singers. If somebody has an interest in "Jennifer Lopez" then one may comfortably presume that this person might buy CDs or movies of Jennifer Lopez. Thus, information about interests can, to some degree, "stand in" for information about products.

We have processed the relational database of demographic and interest information with the WEKA data mining algorithm [13] and have found association rules between classifications of customers and interests. Thus, we needed an ontology that allows us to represent the resulting association rules of a data mining operation in a *succinct* format.

Note that we are not designing a marketing domain ontology which needs to represent all varied aspects of the marketing domain. We are creating an intersection ontology as an integral part of a marketing system. Our application deals with customer classifications needed for a marketing ontology. Our ontology is, in Sowa's terms, an application ontology [14], serving our marketing project [12] described above. As such, our marketing ontology concentrates only on representing purchasing knowledge, as described in detail in Section 2.

The straightforward representation of a customer classification is a tree hierarchy. The root represents the concept PERSON. The various demographic dimensions are ordered. At each of the levels we consider one different demographic dimension according to the above order and branch each node in the previous level to all possible options of this level's dimension. However, as we shall show there are problems with this representation.

To overcome these problems, we draw on Sowa's notion of representing conceptual knowledge using distinctions [15] and on Wille's use of intersections in Formal Concept Analysis [16, 17]. Due to the demands of the domain, realizing there is no natural order among the demographic dimensions and the need for an economical representation, we have developed an ontology that relies heavily on the use of "intersections" of concepts. To further economize, our ontology only contains those intersections about which we have marketing knowledge that needs to be represented. Thus, concepts are inserted into the ontology dynamically on demand. In an intersection hierarchy all the options of all dimensions are children of PERSON. All the relevant customer classifications appear in the next level, each classification as a child of all its options. Such a representation is called a three-level intersection hierarchy. Finally we represent a more economical solution where the customer classifications can be distributed over several levels – the multi-level intersection hierarchy.

Section 2 discusses in more detail why an ontology for marketing knowledge is useful. In Section 3, we will show the design of a customer hierarchy by ordered dimensions and the problems arising from it. Then, in Section 4, we will consider an alternative design for the customer hierarchy by creating "intersections" which results in an intersection ontology. In Section 5, we show the network design of a specific kind of intersection ontology, called multi-level intersection ontology.

The evaluation based on our Web marketing project is described in Section 6. In Section 7, we discuss how our marketing intersection ontology relates to Sowa's knowledge engineering by distinctions. Our conclusions appear in Section 8.

2 Representation of Marketing Knowledge

The essence of our marketing ontology is a collection of buy-relationships from customer classifications to product classifications. The basic facts we need to represent are of the form that a specific classification of customers tends to buy a given product or family of products. For example, "Married women with children buy toys." The challenge is to find a representation of this kind of knowledge in a convenient and economical way that fits into our ontology framework.

The marketing ontology needs to contain two hierarchies, a customer classification hierarchy, in short, *customer hierarchy*, and a product classification hierarchy, in short, *product hierarchy*. The group with the classification MARRIED WOMAN WITH CHILDREN (TOY) needs to be identifiable in the customer (product) hierarchy, either as a node or a group of nodes. To achieve the desired succinct representation, we prefer a single node for the customer classification

concept and a second single node for the product classification concept. We connect those two nodes by a single relationship link with the label "buys", which is an economical representation capturing the desired marketing knowledge for an ontology.

Figure 2 shows a tiny ontology excerpt of four nodes with three "buys" connections. The node WOMAN WITH CHILDREN and its child MARRIED WOMAN WITH CHILDREN belong to the customer hierarchy. The node TOY and its child DOLL belong to the product hierarchy. The three connections are labeled "buys." The "buys" relationship to TOYs is inherited from WOMAN WITH CHILDREN to MARRIED WOMAN WITH CHILDREN. The inherited relationship is a dashed arrow, usually not shown in diagrams, since it can be inferred.

On the other hand, if the customer classification is represented by k nodes ($k > 0$) and the product classification is represented by l nodes ($l > 0$), then up to $k * l$ "buys" relationships are needed to represent the proper marketing knowledge, which is less desirable. Figure 3 represents a tiny part of a customer hierarchy and a product hierarchy. In Figure 3, two nodes are need to represent "men with children" or "electric toys". Thus, 4 arrows are needed to represent the fact that "men with children buy electric toys."

An alternative way with nodes representing "men with children" and "electric toys", respectively, with an arrow connecting them offers a more economical representation. However, if we represent ELECTRIC TOYS and NON-ELECTRIC TOYS at level two and the distinction between OUTDOOR and INDOOR at level three, then "men with children buy outdoor toys" will require 4 arrows. As we will discuss later, for each sequential ordering of the relevant dimensions, there are some marketing knowledge facts with an uneconomical representation.

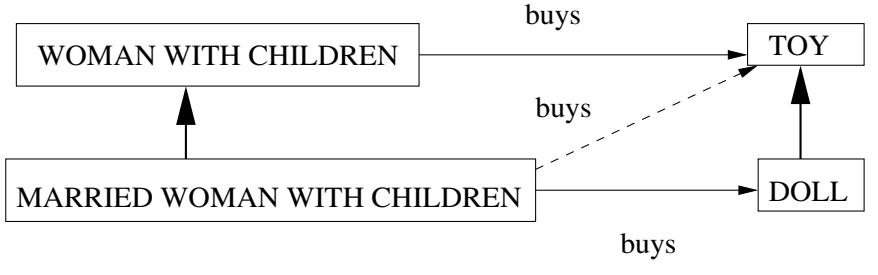


Fig. 2. Extract of a Marketing Ontology

We use the link with the label "buys" to mean "is likely to buy". Thus, "buys" is a statement strictly about a (meaningful) percentage of the population satisfying the demographic data.

For practical usability, a marketing knowledge representation should be as simple as possible. For example, if data mining tells us that married men with children buy diapers, and married women with children buy diapers, then an

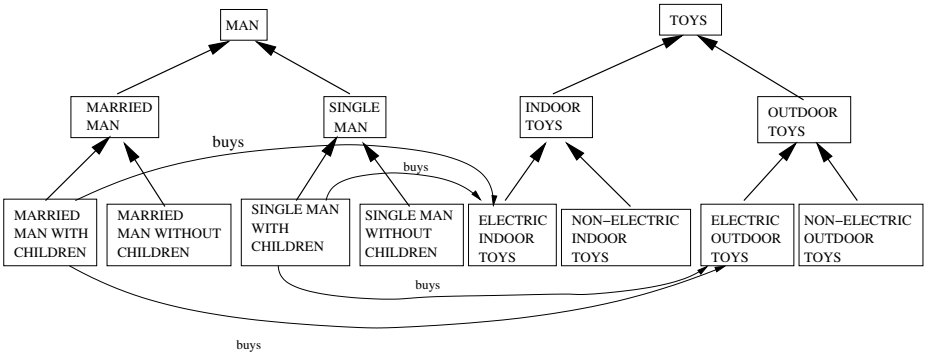


Fig. 3. We need $k * l$ arrows to express a simple Marketing Fact

assertion that married people with children buy diapers is better. Such information should be attached to exactly the concepts about which we are expressing knowledge. In our case, we would like to associate this knowledge with the concept *married people with children*, assuming such a concept *exists* in the ontology.

Finding a marketing ontology that enables the representation of *all the needed concepts* explicitly without creating a combinatorial explosion of concepts for customers is non-trivial. An intersection ontology achieves exactly this goal. However, first, we will describe the straightforward alternative, a tree representation with ordered dimensions, and explain why it is inappropriate for marketing knowledge.

3 Customer Tree Hierarchy with Ordered Dimensions

Following customary practice in marketing, as used, for instance, by MediaMark [18], we perform a classification of customers along various dimensions such as gender (man, woman), age (five age groups), marital status (single, married, separated), children status (with children, no child), *etc.*

Marketing research may reveal knowledge about buying habits of a customer classified according to several dimensions simultaneously. For example, consider the sentence: "Middle-aged married men with children buy books on early childhood development." We want in the customer hierarchy a node which corresponds exactly to the above customer classification.

Consider a tree hierarchy according to the four dimensions listed above, each dimension appears at a different level of the hierarchy. The tree hierarchy starts with the root node PERSON at level 1. The division into the classifications MAN and WOMAN happens at level 2. The division of men (and of women) according to five age groups happens at level 3. There is an obvious redundancy, as the same age choices are made twice, once below MAN and once below WOMAN. The next two levels follow the distinction according to marital status among three options, and children status, respectively. For a figure of a similar tree hierarchy see Figure 6(a) in Section 6.

In this tree hierarchy, which we will refer to as T, we are using a linear order of the various dimensions of a customer. In other words, we prioritize the different dimensions. The above order of dimensions was working well for the above given example, because the customer class (middle-aged married men with children) is represented by a unique leaf node which is the source for the "buys" relationship to the node representing the product BOOKs ON EARLY CHILDHOOD DEVELOPMENT.

Some marketing knowledge should be attached at a single non-leaf node in the tree hierarchy T. For example, "Men buy football tickets" would be expressed by a relationship that has the second level node MAN as its source and FOOTBALL TICKET as its target.

In the last examples, customer classification is represented as one node in T, from which one "buys" relationship link to a product node is emanating. In other situations, the description of a class of customers may not fit so neatly into the tree hierarchy T, as there might be a mismatch between this class and the order of dimensions in T. Consider, "People with children invest in Education IRAs." Even older people may have children, and people may also invest in IRAs for their grandchildren, so no age bracket applies here. To capture this class of customers, we need to refer to 30 leaf nodes in the tree hierarchy T, since the dimension considering children is at the lowest level in T. Furthermore, each of those nodes will require a "buys" relationship to an EDUCATION IRA node in the product hierarchy. The marketing knowledge "People with children invest in Education IRAs," expressed in a short sentence, requires 30 links in our marketing ontology. This is clearly an uneconomical representation of marketing knowledge. However, there is no inherent reason why we chose, for example, the distinction between MAN and WOMAN at the second level, above all the other dimensions. If, for example, the children status dimension would have been chosen as the top-level dimension in the hierarchy, then one node and one "buys" link would have been sufficient to represent this customer class and the associated marketing knowledge. Hence, for every ordering of the dimensions, the hierarchy will be well matched to some customer classes but ill fitting for others. Thus, we have identified a serious problem which may occur for *any* choice of ordering the dimensions, where many cases of marketing knowledge will require many links. The problem is inherent in the fact *that* an ordering is used.

Besides this problem of uneconomical representation of marketing knowledge, this straightforward representation has two secondary problems. One problem is the explosion of the total number of nodes. The number of just the leaves in T is the product of the numbers of options for all dimensions. In our tree hierarchy T of only four dimensions, each with few choices, there are 60 leaves. In the market research field, practitioners have identified many more dimensions. For example, the ten dimensions appearing in the MediaMark Web site [18] for customer classification are: Gender, Age, Household income, Education, Employment status/occupation, Race with region, Marital status, County size, Marketing region, and Household size. Since any combination of dimensions may appear in a customer classification, the tree hierarchy must be fully developed by

expanding all dimensions. This need was demonstrated before with the example using the classification PERSON WITH CHILDREN.

The second problem with ordered dimensions is related to the explosion of nodes. Whole subtrees are repeated over and over. For example, the subtree with the marital choices is repeated for every age group. If a marketing executive decides to add a marital status "WIDOWED", then this update has to be performed in every subtree, leading to the well-known danger of inconsistencies (update anomalies).

4 Customer Intersection Hierarchy

The difficulties we encountered in designing a tree hierarchy customer ontology stem from the fact that there is no preferred order of the various dimensions. Thus, a possible solution is to avoid prioritizing the dimensions. To solve this problem we draw on Sowa's notion of representing conceptual knowledge using distinctions [15]. Sowa claims, for example, that there is no order between the distinctions Concrete/Abstract and Object/Process. All four concepts: Concrete, Abstract, Object, and Process are children of Thing. A concept such as PhysicalObject is an intersection of the concepts Concrete and Object. We call the result of consistently applying such distinctions for all dimensions *on demand* an *intersection ontology.* The significance of creating concepts for an ontology only on demand will be explained below.

We note that we may encounter some dimensions without a natural priority between them in the product hierarchy as well. Figure 3 demonstrates this situation between the location dimension (indoor, outdoor) and the operating mode dimension (electric, non-electric) of toys. Nevertheless, the situation in general is quite different from that of the customer hierarchy, where all dimensions are mutually independent. In the marketing field, there is an established practice of considering some dimensions of product classification prior to others. For example, Men's Wear and Women's Wear are typically in different departments and probably even on different floors of a department store. Each of these are further partitioned into various kinds of clothing, shoes, accessories *etc.* Furthermore, customers are used to this ordering of products and search accordingly for what they desire. Hence, while in the customer hierarchy, all dimensions are independent, some dimensions without natural priority between them exist for products. To handle these cases of independent dimensions for products, one could follow Sowa's [15] practice, where intersections appear only for these few mutually independent dimensions. In the balance of this paper, we will concentrate on the customer hierarchy.

The customer intersection hierarchy has a unique root node representing the concept PERSON at level 1. Each option of each dimension is now represented as a child of the root node at the second level of the hierarchy (see Figure 4). We call such a node an *option node.* For example, in Figure 4, we have the WOMAN option node and the MARRIED option node.

The next question is how to represent a customer classification involving several dimensions. For example, the classification MARRIED WOMAN WITH CHILDREN involves three dimensions: gender, marital status and children status. The solution is to define in the hierarchy a new kind of node that represents a combination of several options, one option for each of several dimensions (Figure 4). For example, a MARRIED WOMAN node represents the combination of the option WOMAN for the gender dimension and the MARRIED option for the marital status dimension. Another node represents WOMAN WITH CHILDREN, a combination of options for gender and children status. The more complicated classification MARRIED WOMAN WITH CHILDREN represents a combination of options for three dimensions: WOMAN for gender, MARRIED for marital status and WITH CHILDREN for children status.

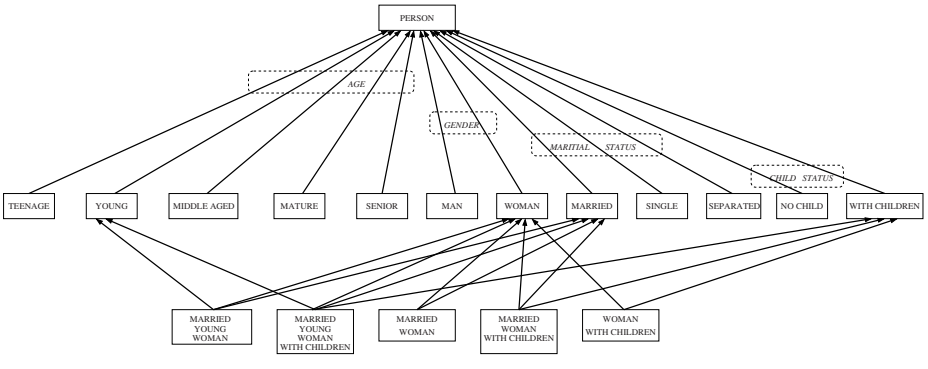


Fig. 4. A Sample Customer Intersection Hierarchy with Option Nodes in Level 2

We call a node that represents a combination of options of various dimensions an *intersection node*, since it represents the classification of a set of customers which is the mathematical intersection of several sets of customers, each with a one-dimensional classification. For example, the set of MARRIED WOMAN is the intersection of two sets MARRIED and WOMAN. Every intersection node is a child of each of the option nodes corresponding to the options involved in the intersection. For example, MARRIED WOMAN is a child of both MARRIED and WOMAN option nodes. Hence all intersection nodes appear in level 3 of the customer intersection hierarchy. Intersection classes of different kinds have appeared in various Object-Oriented Database (OODB) models of medical ontology representations [19–24].

Note that the representation of Figure 4 is superior to the tree hierarchy representation of Section 3, where neither of the classifications mentioned above in this section corresponds to a single node. For instance, MARRIED WOMAN WITH CHILDREN needs to be represented by several nodes in the tree hierarchy T, because the AGE dimension is not mentioned in this classification. In T, AGE is the second dimension, and both MARRIED and WITH CHILDREN are

below AGE in the hierarchy. Thus, to incorporate MARRIED, all AGE choices are included, too. As a result, five nodes of T are needed due to the five options of the AGE dimension. Each of these nodes will have a link to DOLL, to capture the marketing knowledge "Married women with children buy dolls," represented by one link in Figure 2. Hence T is not an economical representation of this marketing knowledge.

As another example, fifteen nodes are needed to represent WOMAN WITH CHILDREN in T. This number corresponds to the multiplication of the number of options for the AGE and MARITAL STATUS dimensions, both not mentioned in this classification. Again, 15 links will be needed to represent the marketing knowledge "Women with children buy toys," represented by one link in Figure 2.

The reason for this large number of nodes of T for a classification is that in the tree hierarchy, for each dimension added to the classification, the number of relevant nodes is multiplied by the number of options for this dimension. This is because at each level the classification of each node of the previous level is further subdivided into nodes according to the options considered at this level. Thus, the representation in Figure 4, using intersection nodes, has the advantage that each classification (selecting one option for each dimension), independent of the number of dimensions involved, and independent of the number of their options, is represented by a single node. Hence, each "buys" link, starting at a customer class that is described by an intersection node, has a unique source. In contrast, in the customer tree hierarchy T, it is typical to require several nodes with "buys" relationships for such a classification.

Option nodes may have attributes and relationships. Intersection nodes inherit these properties from all their parents, enabling multiple inheritance of properties. The root node and option nodes may also be sources in "buys" relationships.

At first glance it might appear that with intersection nodes we will generate hierarchies that are even larger than with ordered dimensions, as we have a large number of nodes already at the second level. However, the opposite is the case. A crucial aspect of our definition of intersection ontologies is that concepts below the second level are only created on demand. That is, *only* nodes which represent a combination of dimensions needed for the marketing knowledge in our database are represented in the hierarchy. If no marketing knowledge about a specific combination of dimensions exists, then we do not create an intersection node for this combination!

More specifically, if we do not need a specific group of customers from our database as the source of a "buys" relationship then we do not need to create the corresponding concept node. Thus, if there is no marketing knowledge available in our database about a single man of Alaskan ethnic origin over seventy, then we will not create a corresponding general concept in our ontology. Intersection nodes are created only on demand if the need for them arises. Traditional general ontologies and domain ontologies typically attempt to represent everything that may exist. For our marketing application, this would result in an explosion of concepts. With the intersection hierarchy, the explosion of nodes is controlled.

Only concepts needed are created. In the ordered dimension representation, a node which is not a leaf cannot be omitted from the tree hierarchy, even if no marketing knowledge is available regarding this node, since marketing knowledge may exist about any of its descendants.

Definition: The size of an ontology is a pair (a, b) where a is the number of nodes and b is the number of relationships.

For instance, the size of the ontology of Figure 4 is (18, 26).

5 Multi-level Intersection Hierarchy

Now we will explicitly consider the network connecting all the nodes in the customer intersection hierarchy. We first describe formally the network of the intersection hierarchy, informally described in the previous section. This network will be denoted the three-level intersection hierarchy. Our discussion will show that the three-level intersection hierarchy is not a proper representation. We will then introduce an alternative network, the multi-level intersection hierarchy, overcoming the deficiencies of the three-level intersection hierarchy.

Consider an intersection node which represents the concept of a combination of k options O_{i_1}, O_{i_2},..., O_{i_k}, one for each of the corresponding k dimensions ($k \leq n$) of the n existing dimensions. Such a concept (node) is more specific than (a child of) each of the option concepts (nodes) which represents one of the options O_{i_j}, $1 \leq j \leq k$, since the set of customers which satisfy all the options O_{i_1}, O_{i_2},..., O_{i_k} simultaneously, is a subset of each of the customer sets which satisfies one option O_{i_j}, where $1 \leq j \leq k$.

In the three-level intersection hierarchy, each intersection node is at the third level, since all its k option parents are at the second level. Hence, the name of this network. (See Figure 4 for a sample of a three-level customer hierarchy.)

Now we will discuss in detail why the three-level intersection hierarchy is improper. In the three-level intersection hierarchy, only IS-A relationships between an intersection node and option node are presented. Consider two specific intersection nodes in Figure 4, MARRIED WOMAN and MARRIED WOMAN WITH CHILDREN. The second classification is more specialized than the first classification, since the set of customers, classified by MARRIED WOMAN WITH CHILDREN, is a subset of the set of customers classified by MARRIED WOMAN. To express this specialization, the intersection node MARRIED WOMAN WITH CHILDREN should have as a parent the intersection node MARRIED WOMAN.

In the three-level intersection hierarchy in Figure 4, the node MARRIED WOMAN WITH CHILDREN has three parents: WOMAN, MARRIED and WITH CHILDREN. Should those parent relationships also exist after adding the parent MARRIED WOMAN? The node MARRIED WOMAN itself has as parents the option nodes WOMAN and MARRIED. A relationship from MARRIED WOMAN WITH CHILDREN to WOMAN (or to MARRIED) is implied by the transitivity of the IS-A relationship.

Thus, we conclude that the three-level representation does not fulfill all our requirements for a proper representation, because it does not capture the specialization which exists between intersection nodes. We will now introduce an alternative representation, the more refined *multi-level intersection hierarchy* that allows expressing parent-child relationships between two intersection nodes, when one represents a more specific concept than the other. For a multi-level hierarchy representation of the nodes of Figure 4, see Figure 5.

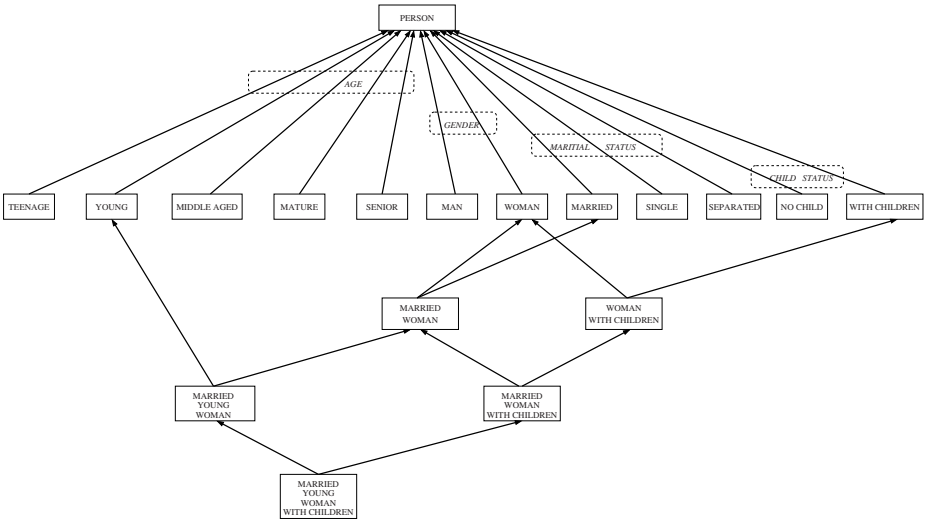


Fig. 5. A Sample Multi-Level Customer Hierarchy

In Figure 5, the node MARRIED WOMAN WITH CHILDREN has no parent relationship to the option nodes. On the other hand, the node MARRIED YOUNG WOMAN has a parent relationship to the option node YOUNG since the hierarchy contains neither the node YOUNG WOMAN nor the node YOUNG MARRIED which would have been parents of MARRIED YOUNG WOMAN and would have implied, as an intermediate node, the IS-A relationship to the option node YOUNG by transitivity.

Note that Figure 5 has 5 levels. The number of explicit parent relationships in Figure 5 is 22 versus 26 such relationships in Figure 4. Both figures have 18 nodes.

The definition for visual complexity of a diagram was introduced and used in [25–27]. We will now modify it for use with ontologies.

Definition: The *visual complexity* C of an ontology of size (a, b) is the ratio of the number of relationships (= links) to the number of nodes, $C = b/a$.

Hence the three-level intersection ontology of Figure 4 has size (18, 26) and visual complexity $C = 26/18 = 1.44$. On the other hand, the multi-level intersection ontology of Figure 5 has size (18, 22) and visual complexity $C = 22/18 = 1.22$.

In this example, the multi-level ontology has lower size and lower visual complexity in comparison with the corresponding three-level ontology. Note that our visual complexity measure is a global measure for an ontology, compared to the notions of "tangled" and "sparse" used in [28] to measure local properties of the top level hierarchy.

To summarize, the three-level intersection hierarchy representation is not proper, because it does not capture IS-A relationships between intersection nodes. Such relationships are captured by the multi-level intersection hierarchy which also has other advantages, as follows.

1. The multi-level representation allows to use inheritance between intersection nodes, which is not possible in the three-level hierarchy. For example, if we know that women with children buy toys, we inherit this fact to married women with children. In this way, the multi-level representation maintains one of the major advantages of ontologies, the economy brought about by inheritance-based reasoning.
2. The distribution of intersection nodes over several levels, due to the additional specialization IS-A relationship between such nodes, simplifies orientation of the user in such a hierarchy.
3. The number of explicit parent relationships is typically smaller than in the three-level intersection hierarchy. (This is not necessarily true, as one can intentionally design a counterexample.) This makes the multi-level intersection hierarchy diagram smaller in *size* and lower in *visual complexity* than the equivalent three-level intersection hierarchy.

6 Evaluation

We will use the customer ontology of our marketing project to evaluate the design of the multi-level ontology versus the other designs. In our Web marketing project, we have collected 301,109 valid records of person's information. A record of information is considered valid when it has a valid email address and at least one expressed interest. Some of the information is expressed in foreign characters, which we ignore. After filtering, we have 274,665 records. However, most people also provide more information such as their age, gender and marital status. Regarding these as three dimensions for PERSON, we constructed the customer ontology for our project and show how the *ordered dimensions* tree hierarchy, the *three-level intersection* hierarchy, and the *multi-level intersection* hierarchy representation will perform, respectively.

The dimensions of AGE, GENDER and MARITAL STATUS have 6, 2 and 6 options respectively. Each record is represented as an instance of a corresponding classification (node) in the ontology. However, some nodes only contain fewer than 100 records. For marketing purposes, we ignore such nodes which do not represent useful information.

Using the design of *ordered dimensions*, we have the ontology as in Figure 6(a). The blank boxes stand for nodes without enough instances, and are not created.

In this figure, each node represents a meaningful customer classification from a marketing point of view, with the corresponding number of persons in our database. For instance, there are 23709 records for those who are males whose ages are between 10 and 19, whose marital status is not specified.

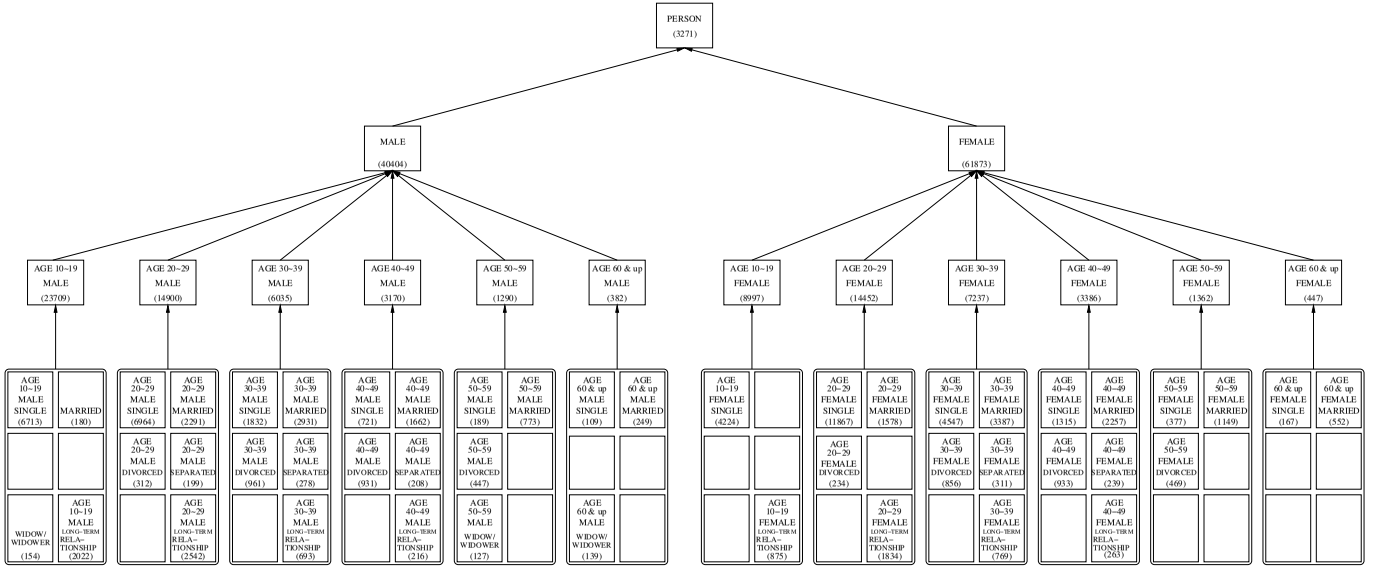
The tree hierarchy in Figure 6(a) has 62 nodes and 61 IS-A links and the visual complexity of 0.98. However, using this hierarchy, when trying to represent all the customer concepts with marketing knowledge, some of the concepts are not represented by a single node. To represent such a concept, multiple nodes, distributed in different parts of the hierarchy of Figure 6(a), have to be collected. For example, due to the order of the dimensions, to represent the concept AGE 20-29, 11 nodes, structured in 2 subtrees in Figure 6(a), are needed, as shown in Figure 7(a). Moreover, to represent the concept MALE and DIVORCED, 4 nodes need to be collected, as shown in Figure 7(b).

The number of possible concepts with one dimension is $2+6+6 = 14$ and with two dimensions is $2\times6+2\times6+6\times6 = 60$. Hence the number of possible concepts with one or two dimensions is 74. We are not considering here the concepts with three dimensions, since they are properly represented in Figure 6(a) by a single node leaf. Among those 74 concepts, 14 can be found, in levels 2 and 3 in Figure 6(a), as corresponding single nodes. Since 48 of them do not have enough instances, there are $74 - 14 - 48 = 12$ concepts which are not represented by a single node. Figure 6(b) summarizes those 12 concepts needed in addition to Figure 6(a) to represent every needed marketing knowledge concept. Every one of these 12 concepts needs to be represented by a group of nodes, distributed in various parts of Figure 6(a), shown as its children as in the Figure 7. For each concept in Figure 6(b), the number of these nodes is listed, adding up to 76 nodes. Note that Figures 7(a) and 7(b) show only the expansions of the first node and the sixth node in Figure 6(b), respectively. Thus, the number of nodes representing all the relevant concepts in the customer tree hierarchy is $62 + 76 = 138$.

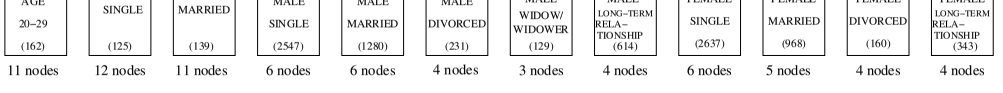
In the design of the *multi-level intersection* hierarchy, we get the ontology hierarchy in Figure 8. There are 14 option nodes. The third level has 21 intersection nodes, each of which has 2 IS-A links to option nodes. The fourth level has 47 intersection nodes combining three dimensions. Out of 72 possible intersection nodes, 25 contain fewer than 100 records and are not represented. Thus, this design has $1 + 14 + 21 + 47 = 83$ nodes and 150 IS-A links. The visual complexity of the *multi-level intersection* hierarchy is $150/83 = 1.81$.

For the *three-level intersection* hierarchy, the figure is too large to be shown here. However, the figure is a modification of Figure 8 for the *multi-level intersection* hierarchy. The only difference, is that all the 47 nodes in the fourth level are moved to level 3 and are directly connected to the option nodes. Thus, we have 68 intersection nodes at level 3. The total nodes number again is 83, but the number of IS-A links is 197. The extra 47 IS-A links are since each of the 47 nodes has 3 IS-A links. The visual complexity is $197/83 = 2.37$.

In summary, the usage of intersection nodes insures that every relevant customer concept is represented by one single node in the hierarchy. The three-level



(a)



(b)

Fig. 6. Our Marketing Hierarchy with Ordered Dimensions

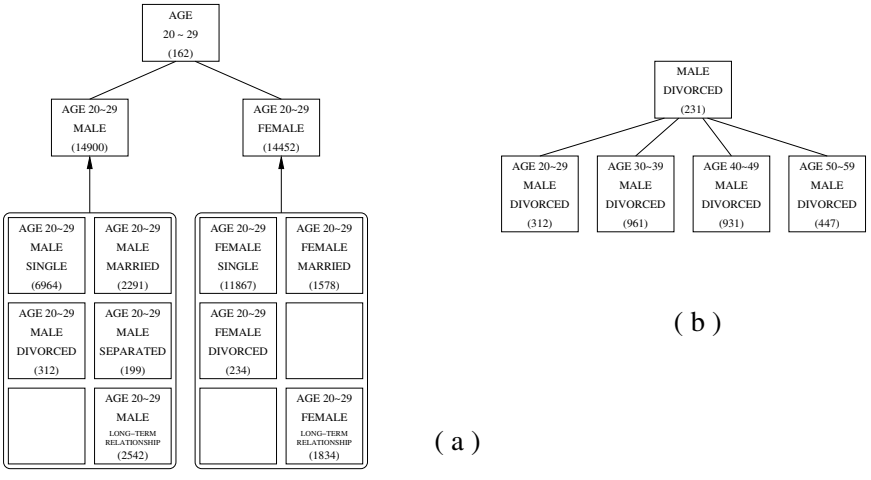


Fig. 7. The nodes collection samples in the Figure 6(b)

and multi-level intersection hierarchy have the same number of nodes. However, the *multi-level intersection* hierarchy has fewer links and lower visual complexity than the *three-level intersection* hierarchy.

In the design of the customer tree hierarchy, consisting of several trees (see Figure 6), there are 138 nodes. Comparing with the 83 nodes in the other two designs, the size is 66% bigger. This design has a lower visual complexity, since it is a forest of 13 trees. However, the measurement of visual complexity is secondary to the size, in this case.

7 Discussion

In principle, the mission of a general purpose ontology is to represent the real world and facilitate the exchange of information between heterogeneous systems. In this view, one would need to create every single intersection in the customer hierarchy, whether we have additional information about it or not, simply because it may be needed for information exchange. However, in our application ontology this would lead to an unreasonably large structure. As a matter of fact, Sowa [14], page 53, writes that "limited ontologies will always be useful for single applications in highly specialized domains. But to share knowledge with other applications, an ontology must be embedded within a more general framework." As a single application ontology, our marketing ontology has to serve its application as well as possible. Thus, it should be an economical representation that includes only nodes for which marketing knowledge is relevant. Also, for this specific application, there is no order whatsoever between any pair of dimensions. Thus, intersections need to be applied universally. This is not necessarily true for other application ontologies.

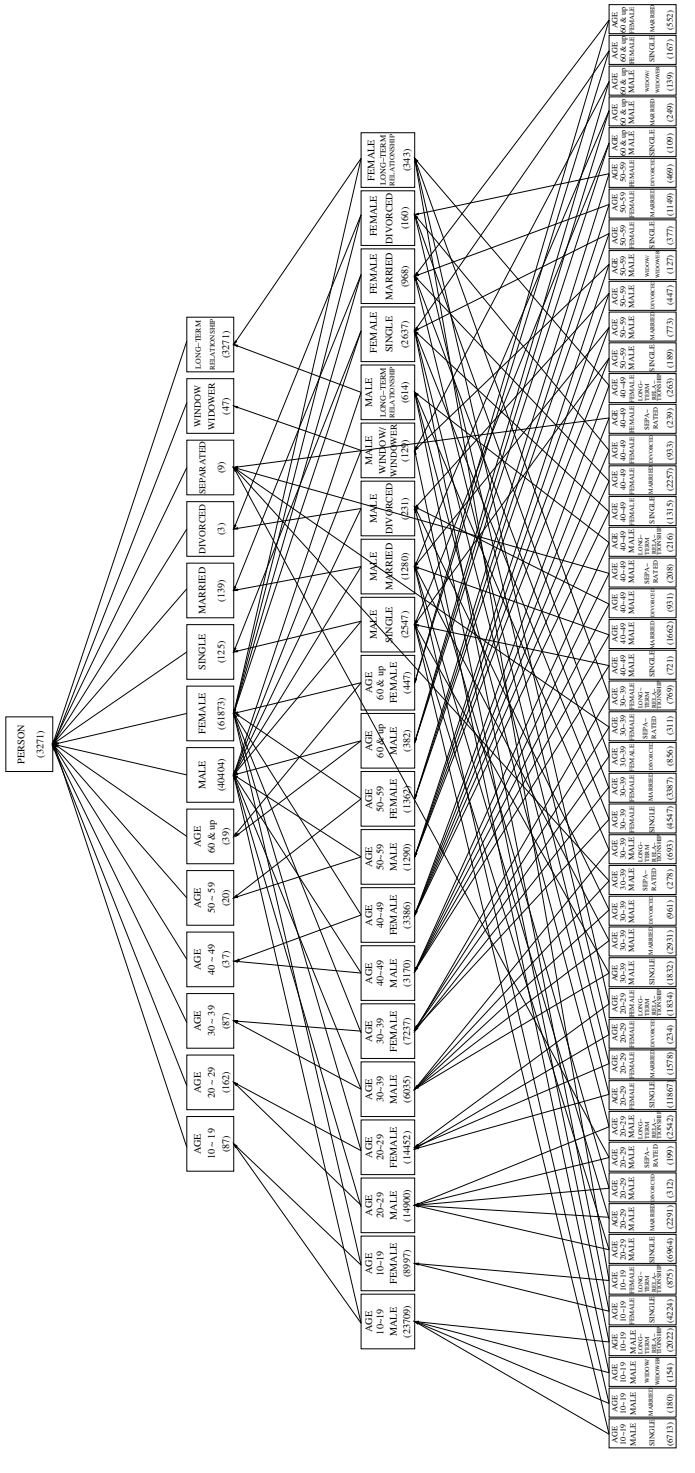


Fig. 8. Our Marketing Multi-Level Customer Hierarchy

Thus, we omit from our ontology information that may be "inferred." The principle of creating an ontology that is an economical representation goes back to the first semantic networks [1]. A major reason for storing attributes at a concept high up in a semantic network hierarchy was to eliminate duplication of information. Whenever any such attribute was needed at a lower level, it was inherited down.

The major difference in our case is that we are not omitting attributes at lower levels, we are omitting the lower levels altogether. Instead of using inheritance to instantiate the representation whenever needed, we are allowing the on demand creation of new intersection concepts which are children of two or more existing concepts. Similarly, [29] writes about dynamic additions in an ontology: "Note that according to this definition, an ontology includes not only the terms that are explicitly defined in it, but also terms that can be inferred using rules." Thus, one could view our approach as implementing a global inference rule which is triggered by existing data and infers new concepts.

One problem which exists in the design of ontologies is how to forbid the representation of an impossible combination. In our intersection ontology design, this translates into the question how to forbid impossible intersections. For example, we should not represent an intersection node TEENAGE MARRIED WOMAN since it is illegal for a teenager to get married.

How can we prevent impossible combinations in our intersection ontology? Note that our ontology's intersections are created on demand, based on available marketing knowledge. There should be no such impossible combinations in the available marketing knowledge, and thus no such intersection should be created. If, however, such an intersection is created, it comes from erroneous data and can be used for auditing errors in the given marketing knowledge, as we did in [23].

8 Conclusions

We have introduced an application-oriented ontology for marketing knowledge, based on the introduction of intersection concepts on demand. Instead of imposing an order on the classification dimensions which is satisfactory for some purposes (and users) but not for others, we completely eliminate ordered dimensions. Instead, we consistently use intersections of options for the various dimensions.

We described the development of an application ontology for customer classifications in a marketing knowledge base. This ontology needed to conform to a number of requirements. First and foremost, we wanted to make it easy to represent in the ontology, knowledge about likely buying behavior of classes of customers by a single (or a few) links from customer concepts to product concepts.

The intersection ontology representation satisfies this purpose because it allows the representation of "buys" relationships by single links whenever this is warranted by the marketing knowledge. Yet, our representation does not produce a combinatorial explosion of all possible intersection nodes. Rather, it only

represents the concepts for customer classes which are necessary as sources for known "buys" relationships.

In the multi-level intersection ontology representation, intersection nodes of many option nodes may be placed at various levels. These nodes may have IS-A links to other intersection nodes as well as to option nodes. These IS-A links may be used for property inheritance, as in other concept hierarchies.

The multi-level hierarchy representation fulfills a secondary requirement that we have for ontologies, namely that they can be represented by diagrams of relatively low size and visual complexity. This representation typically requires lower visual complexity relative to the three-level intersection hierarchy. As described in Section 6, in the evaluation based on our marketing project, the multi-level representation has a 24% lower visual complexity than the three-level representation. Moreover, its size is 40% smaller than the size of the ordered dimensions representation. In conclusion, we showed that for the marketing domain an economical intersection ontology may be created by inserting intersections of options of the various classification dimensions on demand. Such a representation may also be proper for other applications.

References

1. Quillian, M.R.: Semantic memory. In Minsky, M.L., ed.: Semantic Information Processing. The MIT Press, Cambridge, MA (1968) 227–270
2. P. Bouquet, A. Dona, L.S., Zanobini, S.: Contextualized local ontology specification via ctxml. In: MeaN-02 AAAI Workshop on Meaning Negotiation, Edmonton, Alberta, Canada (2002)
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5** (1993) 199–220
4. Geller, J., Perl, Y., Lee, J.: Guest editors' introduction to the special issue on ontologies: Ontology challenges: A thumbnail historical perspective. Knowledge and Information Systems **6** (2004) 375–379
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284** (2001) 34–43
6. Guarino, N., Poli, R.: Special issue on formal ontology in conceptual analysis and knowledge representation. Journal of Human-Computer Studies **43** (1995)
7. Hendler, J.: Special issue on agents and the semantic web. IEEE Intelligent Systems **16** (2001)
8. Gruninger, M., Lee, J.: Special issue on ontology applications and design. Communications of the ACM **45** (2002)
9. McGuinness, D.L.: Ontologies come of age. In Fensel, D., Hendler, J., Lieberman, H., Wahlster, W., eds.: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential, Washington, D.C., MIT Press (2002)
10. Noy, N.F., Hafner, C.D.: The state of the art in ontology design. AI Magazine (Fall 1997) 53–74
11. Geller, J., Perl, Y., Lee, J.: Special issue on ontologies: Ontology challenges. Knowledge and Information Systems **6** (2004)
12. Geller, J., Scherl, R., Perl, Y.: Mining the web for target marketing information. In: Proceedings of CollECTer, Toulouse, France (April 20th, 2002)

13. Witten, I.H., Frank, E.: Data Mining. Morgan Kaufmann Publishers, San Francisco (2000)
14. Sowa, J.F.: Knowledge Representation. Brooks/Cole, Pacific Grove, CA (2000)
15. Sowa, J.: Distinctions, combinations and constraints. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995)
16. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In Rival, I., ed.: Ordered sets, Reidel, Dordrecht, Boston, MA (1982) 445–470
17. Wille, R.: Concept lattices and conceptual knowledge systems. Computers and Mathematics with Application **23** (1992) 493–515
18. MediaMark: MediaMark Web site (2001) http://www.mediamark.com.
19. Gu, H., Halper, M., Geller, J., Perl, Y.: Benefits of an OODB representation for controlled medical terminologies. JAMIA **6** (1999) 283–303
20. Gu, H., Perl, Y., Geller, J., Halper, M., Liu, L., Cimino, J.J.: Representing the UMLS as an OODB: Modeling issues and advantages. JAMIA **7** (2000) 66–80
21. Liu, L., Halper, M., Geller, J., Perl, Y.: Controlled vocabularies in OODBs: Modeling issues and implementation. Distributed and Parallel Databases **7** (1999) 37–65
22. Liu, L., Halper, M., Geller, J., Perl, Y.: Using OODB modeling to partition a vocabulary into structurally and semantically uniform concept groups. IEEE Transactions on Knowledge and Data Engineering **14** (2002) 850–866
23. Geller, J., Gu, H., Perl, Y., Halper, M.: Semantic refinement and error correction in large terminological knowledge bases. Data and Knowledge Engineering **45** (2003) 1–32
24. Gu, H., Perl, Y., Elhanan, G., Min, H., Zhang, L., Peng, Y.: Auditing concept categorizations in the UMLS. Artificial Intelligence in Medicine **31** (2004) 29–44
25. Gu, H., Perl, Y., Halper, M., Geller, J., Kuo, F., Cimino, J.J.: Partitioning an object-oriented terminology schema. Methods in Medical Informatics (2001) 204–212
26. Perl, Y., Geller, J., Gu, H.: Identifying a forest hierarchy in an OODB specialization hierarchy satisfying disciplined modeling. In: Proc. CoopIS'96, Brussels, Belgium (1996) 182–195
27. Gu, H., Perl, Y., Halper, M., Geller, J., Neuhold, E.J.: Contextual partitioning for comprehension of OODB schemas. Knowledge and Information Systems (KAIS) **6** (2004) 315–344
28. Campbell, A., Shapiro, S.: Ontologic mediation: An overview. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada (1995) 16–25
29. Gómez-Pérez, A., Benjamins, V.R.: Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In Benjamins, V., Chandrasekaran, B., Gómez-Pérez, A., Guarino, N., Uschold, M., eds.: Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden (1999) 1.1–1.15