# FACETED METADATA FOR ANNOTATION AND RETRIEVAL OF WEB-BASED INFORMATION

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
for the Degree of
Master of Science
in
Computer Science
University of Regina

By
Yuancheng Liu
Regina, Saskatchewan
March, 2006

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

*Your file Votre référence*
*ISBN: 978-0-494-20221-0*
*Our file Notre référence*
*ISBN: 978-0-494-20221-0*

## UNIVERSITY OF REGINA

## FACULTY OF GRADUATE STUDIES AND RESEARCH

## SUPERVISORY AND EXAMINING COMMITTEE

Yuancheng Liu, candidate for the degree of Master of Science, has presented a thesis titled, ***Faceted Metadata for Annotation and Retrieval of Web-Based Information,*** in an oral examination held on March 17, 2006. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

| | |
|---|---|
| External Examiner: | Dr. Luigi Benedicenti, Faculty of Engineering |
| Supervisor: | Dr. Daryl Hepting, Department of Computer Science |
| Committee Member: | Dr. R. Brien Maguire, Department of Computer Science |
| Committee Member: | Dr. Cory Butz, Department of Computer Science |
| Chair of Defense: | Dr. Chris Oriet, Department of Psychology |

# Abstract

Today, a multitude of information is available online. However, finding suitable information for a particular purpose is increasingly problematic. Recent work has indicated that *faceted metadata* can enable efficient annotation and retrieval of web-based information. Semantic Web technologies provide different approaches to create and encode the faceted metadata. They make it possible to explicitly represent the background and meaning of web resources in a way that enables, for both humans and machines, the sharing of web-based information. This thesis presents the design and implementation of a system using these technologies. The system can enable the user to dynamically annotate the semantics of web-based data sources and retrieve information of interest. This thesis will present the example of a culinary recipe application domain. To avoid empty or nearly-empty result sets, a similarity analysis is introduced for organizing result sets based on semantic distances. A usability study was conducted to evaluate the system with a sample recipe collection. The study found that the interface with facet space cues was useful for browsing and annotating the data collection.

# Acknowledgements

I would like to thank my advisor, Dr. Daryl Hepting, for his support, guidance, and patience. His insight and knowledge helped motivate me and his direction kept me on track while writing my thesis.

I would like to express my thanks to Dr. Brien Maguire and Dr. Cory Butz, as part of my thesis committee, for their valuable suggestions and guidance to improve my work.

I would like to thank Mr. Tim Maciag for his kind help and Mr. Alan Friesen for his proofreading of the final document.

I would like to thank all the people from the entire *cogito* project team for their help in one way or another.

I gratefully acknowledge that my studies were funded in part by the Faculty of Graduate Studies and Research, the Department of Computer Science, and the Natural Sciences and Engineering Research Council of Canada through a Discovery Grant to my supervisor.

Lastly, I would like to thank my husband Peter Huang and my son Harry Huang for their patience and support over the years.

# Post Defense Acknowledgements

I thank my external examiner, Dr.Luigi Benedicenti, for his interest and many valuable comments.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Today the World Wide Web is a huge, loosely organized library of information providing remote access to more than 3.3 billion multimedia documents [45]. This growing amount of information means that, over times finding suitable information for a particular purpose will become more complicated. A usability study [38] of 69 web sites showed that of all site searches, 53% ran into trouble due to poorly organized search results and 32% were affected by poor information architecture. This situation presents the challenge of how to effectively describe and manage the semantic information of the web for both human and computer. Ideally, the computer should provide good support for the search or browse task while retrieving all relevant information of interest. Recent work has indicated that *metadata* can enable efficient annotation and retrieval of web-based information [28, 32, 41, 44, 49]. Metadata provides basis for extension of the *cogito* system to handle web-based applications.

The main contribution of this work lies in the design and implementation of a metadata-based system to annotate web documents with semantic information and to enable satisfying exploration of that information by adapting the *cogito* system. Several methods are examined in the context of the culinary recipe application domain which helps to illustrate the processes and principles involved, but they are also applicable to any web-based application domain. The recipe domain was chosen because of its wide appeal. The meaning of recipes are well-known to most people. Its semantic description may contain a large quantity of entities and relations. The implemented system should be able to handle such sets of recipes in an intelligent way.

## 1.1 Motivation

The following scenario is intended to represent a fairly common problem, with no easy solution presently available.

> *Jessie plans to attend a party on Friday night. It is now Wednesday evening. She has worked very hard on her thesis and needs some time off. She has been asked to bring an appetizer. She has some beef in her fridge that she would like to use. She needs some ideas on what to make using other ingredients she also has on-hand, since she will not have a chance to go shopping. She is a good cook, she relies on her frying pan and crockpot.*

Due to the number of recipe collections available online, Jessie may access the web to look for *recipes for an appetizer with beef* and find cooking tips. She accesses the most familiar web search engine, Google, in which she enters the query "beef appetizer."[1] The results screen tells her that there are about 1,010,000 results for her query and she starts examining the top 10 results. The first site[2] on the list gives her 13 recipes, each of which she examines by following a hierarchy of concepts and then rejects. She thinks that she could spend a lot of time here and find nothing. She decides to try some new queries. First she tries "beef appetizer fried," since she might be able to use her frying pan to prepare it. She notices immediately that the first result for this query is titled "Appetizer Cheese Bites Recipe - Fried Cheese Bites"[3] which, upon examination, does not contain any beef. Furthermore, after 3 recipes, the remaining 7 of the top 10 are links for restaurant menus. She then tries the query "beef appetizer crockpot" with the idea that she has enough time to use her slow-cooker to prepare the appetizer. The first result for this query is titled "Crockpot Sausage Recipes"[4] and she is immediately discouraged. Better recipe results might

[1] The query `http://www.google.ca/search?q=beef+appetizer` was performed May 26, 2005. In practice, the "AND" operator is unnecessary and Google includes all search terms by default.

[2] `http://www.thatsmyhome.com/mainstreetdeli/beefaps.htm`

[3] `http://southernfood.about.com/od/cheesesnacks/r/bl30424e.htm`

[4] `http://southernfood.about.com/od/crockpotsausagerecipes/`

be reached if she had used more specific input or employed some advanced search features.

After Jessie's experience on the web, two problems are evident:

- Keyword-based searches match words indiscriminately so the quality of results is not guaranteed. Most queries return thousands of documents including many that are not relevant. Keyword-based search methods do not access the underlying concepts and cannot determine the usefulness of information. It is hard to know whether a recipe, for example, has beef as a significant ingredient. Since keywords can have multiple meanings, irrelevant web pages might contain the keywords while relevant web pages might only contain synonyms for the keywords [6, 38]. The web page creator and its users may often differ on the language they use to describe web content.

- Keyword-based search does not support browsing or exploration, and it provides little or no guidance about how to filter and find a desired result. A user may be looking for a recipe with a specific ingredient such as beef. Sometimes the user needs information in other areas, such as cuisine or how the finished-dish looks. Even if the user finds a starting web page for a recipe collection, in most cases the collection cannot be easily explored because recipes must be examined sequentially within a browser and it is difficult to move back and forth between pages.

## 1.2 Requirements

From the difficulties that Jessie experienced with basic web-based keyword searches, there are some basic requirements evident for an improved recipe search facility:

- Build recipe annotations relevant to the user interest. The meaning and properties of recipes can be important for a user who is not familiar with the recipe. Due to many independent recipe resources available online, recipe data is structured in various ways. The use of metadata standards is a basic requirement for connecting to and communicating with these information resources. Recipe metadata can describe, for example, the type of ingredients and their semantic relations, the course in which the dish is served, and its ethnic origin.

- Support processing of recipe resources for retrieval. To provide a much better retrieval environment, especially when the user does not have in mind an explicit target to be retrieved but would like to browse and explore, recipes of interest have to be retrieved according to content-based retrieval decision mechanisms and according to semantics that the user can understand.

*Metadata* contains relevant data and structure that is flexible enough to be adaptable for different areas of application. It can help the user organize, manage and use information resources in a way that enables machine execution of searches using term suggestion [49], query expansion [27] and flexible matching [30, 32] at the semantic level. Once the recipe data is properly annotated, a user could then instruct a software to find a recipe for an `Appetizer` course with ingredient `Beef` and report the recipe's title, cooking method, and nutrition information.

To reap concrete benefits from adopting metadata, our goal is to organize the semantics of multimedia documents into orthogonal hierarchies and make them available through an alternative search and browsing environment. The work presented here addresses research done concerning these two requirements by applying semantic metadata, which has led to the design and implementation of a new recipe management tool.

This work designs and implements a tool to support the development of sound local practices for annotating, managing, and retrieving recipes. The tool is understood as an application to handle the metadata that provides an overview of the semantic description as a whole. The user should be allowed to interact with this semantic metadata.

A semantic description of the recipe domain can be stored in standard formats like XFML [17] and RDF/RDFS [4, 10], which provide sufficient semantic and structural information. Concrete instances of recipes, as well as the abstract concepts of recipes, should be described.

The semantic description may contain a large quantity of entities and relations. The implemented system should be able to handle such sets of recipes in an intelligent way. The system should be platform independent and extensible. The web is the ideal testing location for this application.

## 1.3 Thesis Outline

This thesis describes the design and implementation of a system for the management of recipes. Chapter 2 gives an overview of faceted metadata and related technologies. It describes the variety of ontology-based approaches for information annotation and retrieval. Related technology and projects will be introduced. Chapter 3 gives details about the design and implementation of the system. Chapter 4 describes the experiments conducted to evaluate the effectiveness of the interface. Finally, Chapter 5 gives a conclusion and suggests possible directions for future research.

# Chapter 2

# Background

Chapter 1 introduced the inadequacy of search engines in locating quality information resources and organizing those results in a useful manner. Currently, various techniques are proposed to support users in searching for relevant information of interest. Among them *metadata*, in its various forms, is one of most common means to improve the quality of search results.

This chapter first presents an overview of metadata and its related technologies. An overview of approaches for improving search results are then presented, followed by a discussion of related projects. Finally, all techniques are reviewed.

## 2.1 Metadata

Metadata is literally defined as "data about data." Some use metadata to refer to machine understandable information for the web [4], while others use it only for records that describe resources at some level of aggregation [2]. Metadata contains relevant data and a structure that is flexible enough to be adaptable for different areas of an application. For example, a library catalogue holds metadata that typically includes the originator of a work, its title, when and where it was published, and the subject areas that it covers. This metadata is commonly linked to the work in the library collection through a call number. Metadata aids the user in finding works about a specific topic with a specific title or from a specific publisher. Web catalogues, such as Yahoo! (`http://www.yahoo.com/`) or Open Directory (`http://dmoz.org/`),

```
<html>
    <head>
        <meta name="title" content="Party Rolls">
        <meta name="description" content=
                    "Quality kitchenware and recipes for cooks.">
        <meta name="keywords" content=
                    "recipes, cooking, outdoor grilling">
    </head>
    <body>
    ...
    </body>
</html>
```

Figure 2.1: Metadata embedded in HTML document.

are also good examples of metadata applications.

Metadata can be kept with the object it describes, or stored separately. For example, as shown in Figure 2.1, metadata is embedded in the `<meta>` tag of the HTML document, giving semantics about a web page such as the title of the page or a list of keywords. Crawler-based search engines can automatically extract and index these structured data. However, most search engines do not trust the `meta` tag's content because of the potential for abuse, such as repeating keywords to boost a site's ranking in search results. As illustrated in Figure 2.2, metadata can also be stored separately with a reference in the `<head>` section of the HTML document through a `link` tag. It simplifies the management of the metadata and makes it possible to have a structured repository for facilitating annotation and retrieval.

### 2.1.1 Applications

Metadata can support the organization, management and use information resources. The following gives a brief list of applications of metadata [2]:

```
<html>
   <head>
      <title>Party Rolls</title>
      <link rel="metadata" href="recipe.meta">
   </head>
   <body>
   ...
   </body>
</html>
```

Figure 2.2: Metadata stored separately.

**Resource Discovery:** Metadata enables effective search of resources across multiple repositories.

**Organizing Resource:** Metadata can organize web-based resources based on audience or topic. These resources can be built dynamically from metadata stored in databases. Various software tools can be used to automatically extract and reformat the information for web application.

**Use Facilitation:** The use of a certain object by different communities can be facilitated by the existence of different metadata records describing it, according to metadata schemes tailored to the needs of each community.

**Interoperability:** Interoperability is the ability to exchange data across different data structures, hardware and software platforms, and interfaces with minimal loss of content and functionality. Structural metadata explaining the semantics of data stored in different sources enable interoperability at the semantic level by solving heterogeneity problems such as having the same name for different kinds of data in different repositories. If the metadata explaining different sources was not created with the same basic vocabulary, then mappings between metadata repositories would also be needed.

**Preservation:** Since digital information is fragile, it can be corrupted, altered or lost during changes of environment. Metadata is crucial to ensure that resources will survive and continue to be accessible in the future by describing how a digital information object is created and maintained, how it behaves, and how it relates to other information objects.

To enable the realization of these metadata uses, various metadata standards and structures have been developed. The well-known metadata vocabulary Dublin Core (DC) [15] provides 15 standard metadata elements for describing resources, including *title, subject and keywords, creator, description, publisher, contributor, date, resource type, format, resource identifier, source, language, relation, coverage and rights management.* This small set of metadata was intended to be used by any community to describe and search across a wide variety of information resources on the web, and attempted to improve searching on the web. However, DC is a very limited way to describe resources. For example, consider a web site for a cooking practice. A typical recipe web page might contain the recipe name, description, ingredients, cooking method, and nutrition information. Without a standard format for recipe storage, the machines cannot construct these concepts. The user would have to manually inspect the information. Machine-understandable metadata is required to convey these concepts at the semantic level. Metadata approaches rely on a predefined set of keywords which are derived by experts and are designed to best describe or represent concepts relevant to the recipe domain. DC can only describe introductory material of the web page, most likely including title, description, author, date, as well as more specialized keywords like `cooking` and `grill`. DC does not have the vocabulary to describe the content of the recipe page, such as using the cook method `grill`. Clearly, DC is not well-suited to extensive content-oriented annotations. Richer metadata is needed to describe the content of web-based resources based on human analysis.

### 2.1.2 Metadata Structure

In order to present information and make it easy to find, metadata organizes web-based resources according to a classification scheme [46, 49].

Like metadata used in Yahoo!, hierarchical metadata uses a hierarchy of human-generated subject categories and site labels to offer a web-wide lookup facility. These categories can be considered as hierarchical classification schemes like that of the Library of Congress or DDC (Dewey Decimal System) [46]. They turn out to be useful for browsing and retrieval when the user knows the path to an object of interest. For example, if the user wants to classify the recipes with `course appetizer` and `ingredient beef`, she can use the hierarchical subject categories of the Yahoo! search engine and try to find the predefined concepts that best describe the recipes. The user must first know to find recipes at `http://health.yahoo.com/recipe/`. There is no direct link there from `http://www.yahoo.com/`. Once at the recipe page, the user can first select the `appetizer` category to restrict the scope of the search. However, this user must then manually search these appetizer recipes to find one with beef. Such hierarchical metadata can not be considered as being "complete" for support of browsing because each resource has a single home and only one path leads the user there. It is not possible to refine an existing search by adding categories so one must start new searches from the beginning. Moreover, the different types of information in one big hierarchy can never address all possible information needed because once established, the hierarchy cannot be reorganized without rebuilding or creating a new one. These hierarchies either become very complicated or rarely provide an effective information retrieval mechanism. It can be difficult to maintain such a scheme to keep coherence and consistency across the relationships that it contains.

Instead of hierarchical metadata, the metadata has several facets: attributes in various orthogonal sets of categories [49]. This is often stored in database record fields and tables. Each facet consists of a set of values describing a domain from a particular aspect. For example, in the recipe domain, possible facets (values) might include "course (appetizer, main dish, . . . )," "cuisine (African, Asia, American, . . . )," "ingredient(beef, beans, onion, . . . )," and "cooking method (fry, cook, . . . )." Facets, like `course` and `ingredient`, are mutually exclusive; a value of type of `course` can never be the value of `ingredient`. This faceted metadata can be seen as a faceted classification scheme [36, 46], which allows the user to find objects of interest through the user's choices by associating each object with zero, one, or more values from each facet. Following a promising path from one facet to another, a user can see exactly the options available at any time and switch easily between searching and browsing. To classify the recipes in this example, the user can select the value from each facet that best describes each of the concepts in the recipes. `Beef` is selected from the `ingredient` facet, and `appetizer` from `course` facet. The user can determine the order in which the facets are selected and explore a large collection through her classification choices until she has a manageable set of recipes to browse. This process is dynamic and scalable. Therefore, the structure of faceted metadata allows users to be quickly educated about relevant concepts that apply to the site's content and supports search tasks in a specific application. Tzitzikas [46] stated that faceted classification is superior to hierarchical classification with regard to comprehensibility, storage requirements, and scalability.

Metadata that forms a semantic structure can be viewed as an ontology. An ontology is a formal, explicit specification of a shared conceptualization [12], which provides a mechanism to create the necessary concepts and properties for describing the content of resources on the web. Ontology is a term borrowed from philosophy which refers to computer taxonomies that specify the logical structure of a controlled vocabulary. For example, consider again a cooking web site. Metadata, like `ingredient` and `method`, do not have any explicit meaning until concepts like *Ingredient* and *Method* are abstracted. All concepts have properties and relations. For example, a recipe document could be connected to a specific ingredient by the relationship "has ingredient," or connected to a person by the relationship "created by." This person may then connect to other recipe documents by the relationship "author of," to an organization by the relationship "has show on Food TV." This can lead directly from one document to others written by the same person, or by others who have cooking shows on the same network. The machine-understandable metadata also helps the user to find a recipe that serves as an appetizer course with ingredient beef and report the recipe's title, cooking method, and nutrition information.

The next section introduces Semantic Web languages that create and use metadata in a semantic way.

### 2.1.3 Languages for Semantic Description

The development of the Semantic Web is tied to ontologies. Berners-Lee describes his vision of a Semantic Web as follows [6]:

> The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. To date, the Web has developed most rapidly as a medium of documents for people rather than for data and information that can be processed automatically. The Semantic Web aims to make up for this.

Semantic Web technologies have made it feasible to add meaning, or semantics, to any resource. It enables sharing information on the web and helps software developers to build applications which can use the semantic description to provide better search environments so that a semantic agent does not have to strip the formatting and pictures from a web page to guess relevant information. Table 2.1 shows various ontology-based languages recommended by the World Wide Web Consortium (W3C).

Currently, ontologies created in XML-based languages are easier to keep open. Resource Description Framework (RDF) is the foundation for process metadata which helps give structure to web content. RDF Schema, or RDFS, is a simple ontology language written in RDF that allows the creation of vocabularies with classes, properties, and class hierarchies. DAML+OIL is an extension of RDF Schema that allows finer-grained control of classes and properties with features such as cardinality constraints and inverses of properties. OWL is a revision of DAML+OIL.

XFML(eXchangeable Faceted Metadata Language) is not a standard language recommended by W3C, but is an enabling technology for the Semantic Web. It is a specialized and simple XML format for exchanging faceted metadata. Its capabilities are, with some restrictions, similar to those supported by the RDF(S) language. The greatest difference present between XFML and RDF(S) for semantic description is in generality: the domain of XFML is restricted and is aimed at generic metadata formats, whereas RDF(S) can be built for any particular domain. Any XFML document can be expressed as RDF, but not the other way around. Both RDF(S) and

| Language | Description |
|---|---|
| XML [9] | provides syntax for structured documents, but imposes no semantic constraints on the meaning of these documents. |
| XML Schema [20] | used to restrict the structure of XML documents |
| RDF [4] | provides a semantic framework to describe resources |
| RDF Schema [10] | used to define vocabularies for describing classes and properties of RDF resources. |
| DAML+OIL [14] | means DARPA Agent Markup Language + Ontology Inference Layer, an extension of RDF Schema language to add more vocabulary for describing properties and classes with features such as cardinality constraints and inverses of properties, relations between classes(e.g.disjointness) |
| OWL [3] | means Ontology Web Language, revision of the DAML+OIL, including cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes |

Table 2.1: Ontology-based languages for semantic description. Explanations of these standard languages can be found in formal recommendations found at http://www.w3c.org.

XFML make the machine representations of resources more closely resemble their intended real world counterparts, and can both be used to define faceted metadata.

This section gives an overview of metadata standards with a focus on XML, RDF(S), and XFML languages for the semantic description.

### XML (eXtensible Markup Language)

XML [9] is a subset of SGML (Standard Generalized Markup Language). While information on the web encoded in HTML focuses on how data is displayed for human consumption, XML as a popular language provides simple, flexible capabilities for storing and exchanging data between machines. The most important benefit of XML is its simplicity. XML documents are human readable, easy to understand and easy to create even in the simplest text editor.

```
<?xml version="1.0"?>
<!  -- A document type (DTD) for recipe example -- >
<!  DOCTYPE Recipe [
    <!  ELEMENT Recipe ( Course | Ingredient)* >
    <!  ATTLIST Recipe ID : CDATA #REQUIRED>
    <!  ATTLIST Recipe URL : CDATA #REQUIRED >
    <!  ELEMENT Course (#PCDATA) >
    <!  ELEMENT Ingredient (#PCDATA)* >
]
<!  -- the recipe data corresponding to its DTD -- >
<!  -- id is a required attribute -- >
<recipe id="4600" url="http://www.cooking.com/recipes/static/">
    <course>Appetizer</course>
    <ingredient>Beef</ingredient>
</recipe>
```

Figure 2.3: A sample XML document.

A well-formed XML document is a labelled tree of elements, where each element corresponding to a tree node has an element type name (called tag name) and a set of attributes with name and value. XML is actually a data format that allows the user to specify her own tags, attributes and data structure, and to predict what type of information might be between tags. The Document Type Definition (DTD) or an XML Schema defines a grammar for the XML documents, which provides constraints on which tags to use and how they should be nested within a document, such as the names of the elements and attributes and their use in documents. Figure 2.3 gives a sample XML document to describe an appetizer recipe with beef as the main ingredient. Tags, like `course` and `ingredient`, carry some semantic information that a machine can understand. Although XML allows the user to add structure to her documents, it says nothing about what the structure means, so the use of XML as a semantic language leaves much to be desired [9].

*XML Namespace* [8] is one of the most important techniques in XML. An XML

```
<recipe xmlns:another="http://another-sample/recipe#"
        xmlns="http://sample/recipe#">
     <ingredient>beef</ingredient>
     <!-- ingredient from another xml document-->
     <another:ingredient>onions</another:ingredient>
</recipe>
```

Table 2.2: An example of XML namespaces.

namespace is a collection of names identified by a URI (Uniform Resource Identifier) [43] reference used in XML documents as element types and attribute names. Namespaces simply distinguish similar names used within the XML documents when these names are properly assigned a prefix (such as `ns-prefix:`) that indicates from which namespace each element or attribute comes. For example, the XML recipe document `http://sample/recipe#` (see Table 2.2) uses the tag `ingredient` in reference to `beef`. Since `onions` is tagged with `another:ingredient`, the definition for ingredient comes from a different file, namely `http://another-sample/recipe#`.

XML namespace is widely used in ontology-based languages. It is intended to define metadata standards on the Web. A namespace schema is a set of metadata element definitions that stand on the Web as reference points to be used to create metadata descriptions about resources of a specific domain in a standardized way. Generally, a namespace schema is designed for a registration authority, and is maintained as a stable reference on the Web. Such a design uses a minimum set of elements with simple structure in order to facilitate the adoption of the schema by communities of users. For example, a namespace schema allows references to different metadata models by the namespace prefix. The vocabulary used in the Dublin Core can be referenced by the prefix `dc`, as in `xmlns:dc="http://purl.org/dc/elements/1.1#."`

### XFML(eXchangeable Faceted Metadata Language)

XFML [17], developed by Van Dijck (`http://petervandijck.net`), is an open XML format for creating and sharing faceted metadata. XFML core is a stable and frozen standard, which means that users can safely build applications which use it without needing to worry about their programs becoming broken when this core is updated. XFML is a model to express the concept of directly connecting topics. Its specification (`http://www.xfml.org/spec/1.0.html`) gives instructions on how to process metadata:

- XFML lets the user exchange faceted metadata. It also lets the user build connections between different XFML maps, by indicating that a topic in one map is equal to a topic in another map.
- XFML expresses a set of concepts (i.e., a conceptual model) in an XML format. It also gives a set of processing instructions that explain how applications should work with XFML data.
- XFML lets the user reuse indexing efforts by publishing metadata, which means the user does not have to index the entire web, but can reuse parts of other XFML maps.
- XFML also provides a simple format to create faceted metadata. Each user can define her own facets and facet values. The data they describe is often kept in a database and published as an XFML document.

An XFML document is a valid XML document and conforms to the XFML DTD (XFML Document Type Definition, `http://xfml.org/spec/xfml.dtd`). Figure 2.4 shows a sample XFML document to describe a facet `ingredient` and its value `beef`.

As shown in Figure 2.4, an XFML map consists of: map information, a set of facets, facet values (called topics here), and a list of pages with occurrences of topics.

**Map Information:** An optional element which describes information about XFML maps, including administrative metadata and connections about the map, such as version, language, editor, publisher, generator, url(where the map is kept, so

```
<?xml version="1.0"?>
<!DOCTYPE xfml SYSTEM "xfml.dtd">
<xfml version="1.0" url="http://cogito/example.xml"
     language="en-us">

<!-- MAPINFO -->
<mapInfo>
<publisher>
     <name>Cooking.com</name>
     <url>www.cooking.com</url>
</publisher>
</mapInfo>

<!-- FACETS -->
<facet id="ingredient">main ingredient</facet>

<!-- TOPICS -->
<topic id="beef" facetid="ingredient">
     <name>Beef</name>
</topic>

<!-- PAGES -->
<page url="http://www.cooking.com/recipes/static/recipe4600.htm">
     <description>A recipe page with defined concepts</description>
     <title>Party Rolls</title>
     <occurrence topicid="beef"/> <!-- Ingredient facet -->
</xfml>
```

Figure 2.4: A sample XFML document.

the user has an unambiguous pointer to topics in the map), and connect(reusing indexing efforts and allowing users to create a web of loosely distributed metadata, connections between two topics in different maps.). For example, Figure 2.4 defines a map information describing a publisher for a given recipe page.

**Facets:** Describes mutually exclusive attributes for a given web resources, e.g., facets like `course, cuisine ingredient`, and `method` for a typical recipe domain. A facet can be defined by tag `<facet></facet>` with `id` attribute, where `id` will be the name used internally to identify the facet. The value is the name of the facet. Figure 2.4 defines the facet `ingredient`.

**Topics:** Describes subjects which only depend on the website we choose. Each topic is the value of a facet. Each topic refers back to its parent facet, defined by `<topic></topic>` tags, with `id, facetid` attributes and other child elements such as name, connect, description, etc. Figure 2.4 defines topic `beef` for facet `ingredient`.

**Pages:** Describes topics that occur on the given page. Adding pages to an XFML document makes it possible to share the user's indexing so other people can reuse it. Pages can be defined by tags <page></page>, with attribute `url`, and with other elements like `title`, `description`, and `occurrence`. Figure 2.4 describes a recipe page with ingredient `beef` and serving as an `appetizer` course.

After an XFML document is created, the user can simply put it on the web as an XML document. This allows web browsers to use XML formatting conventions to display an XFML document and let other people can share it. Alternatively the user can import it in a variety of faceted metadata browsing applications that support XFML.

XFML is a very specific and focused format that only expresses the metadata in the form of orthogonal facets, without trying to syndicate all users' metadata. It is also impossible to specify the cardinality of relations.

| | |
|---|---|
| Resource | `http://www.cooking.com/recipes/static/recipe4600.htm` |
| Property | `http://localhost/cogito/schema/recipe/#hasIngredient` |
| Value | beef |

Table 2.3: A sample RDF statement.

### RDF/RDF Schema

RDF (Resource Description Framework) [4] is a foundation for processing metadata. It is an application of XML. RDF can be used in a variety of application areas, such as resource discovery, cataloguing, and knowledge sharing and exchange, and enable automated processing of web resources. RDF Schema is a simple ontology language expressed in machine-readable format, allowing the understanding of semantic relations among heterogeneous and distributed resources in the web [34]. Based on technologies recommended by the W3C, such as XML and URI, RDF/RDF Schema allows the creation of semantic metadata vocabularies with classes, properties, and relationships between them.

Basically, RDF is a domain-independent model that specifies a value for a property of a resource. RDF describes resources by making statements about resources in a triple form `resource - property(attribute) - value`, where a resource is anything identified by a URI reference; a property is an attribute or relation used to describe resources; and value is literal(such as string) or another resource represented by a URI (called "reification"). A property enables us to clarify meaning and/or constraints on the resource and its value. For example, Table 2.3 specifies "a recipe has ingredient beef."

The resource and its value are defined and related via the property `hasIngredient`, which expresses that the recipe contains a specific ingredient. In fact, there may be many recipes containing ingredient `beef`, but there is only one recipe represented by the specified URI (`recipe:id="4600"`). Therefore, RDF statements remove the ambiguity. Searching for a recipe about ingredient `beef` would look for RDF statements with the specified URI as a resource and beef as value.

RDF helps to give structure to the resource which it describes, but RDF alone

| Name | Comment |
|---|---|
| `rdfs:Resource` | describes everything, i.e.,web page, recipe, image |
| `rdfs:Class` | used to define concepts, such as facet Cuisine |
| `rdf:Property` | used to characterize those classes of things. |
| `rdfs:Literal` | specifies literal values, e.g.,textual strings |
| `rdfs:Datatype` | specifies the rdf datatypes |
| `rdf:Bag` | unordered containers |
| `rdf:Seq` | ordered containers |
| `rdf:Alt` | alternative containers |
| `rdfs:subClassOf` | the subject is a subclass of a class |
| `rdf:subPropertyOf` | the subject is a subproperty of a property. defines a specialization relationship between properties (similar to rdf:subClassOf) |
| `rdf:type` | the subject is an instance of a class |
| `rdfs:domain` | specifies the valid subjects of the statement |
| `rdfs:range` | specifies the valid object values(i.e. beef) |
| `rdfs:comment` | allows one to supply a human-readable comment about the resource |
| `rdfs:label` | provides a human-readable version of a resource's name |
| `rdfs:seeAlso` | provides further information about the subject resource. |
| `rdfs:isDefinedBy` | contains a pointer to a definition of a resource. it is a subproperty of rdfs:seeAlso. |

Table 2.4: An overview of the main vocabulary of RDF(S).

does not sufficiently define the semantics of the application domain, such as what kind of resource is recipe, and how the **hasIngredient** property is related to it.

RDF Schema [10], known as RDF Vocabulary Description Language, offers extensible modelling primitives to define vocabularies that are used in RDF metadata descriptions. Those primitives are a set of RDF resources that can be used to describe RDF resources, including classes, properties and relationships between them. Table 2.4 presents an overview of the main vocabulary of RDF Schema [10].

RDF Schema provides the core classes which are fundamental to the creation of any particular schema: `rdfs:Resource`, `rdfs:Class`, and `rdfs:Property` . Any new class or property defined in a specific RDF Schema must be an instance of them. A root node class `rdfs:Resource` has two subclasses: `rdfs:Class` and `rdf:Property`.

RDF Schema identifies the objects as classes that are organized into conceptual hierarchies. It has the standard semantics of inheritance relationship (`isa`) in object-oriented data models. A class is always a resource(`rdfs:Resource`) whose property `rdf:type` has the value `rdfs:Class`. Properties in RDF Schema are used to denote attributes. Unlike the object-oriented paradigm, properties in RDF Schema are not defined within the scope of a class, but within a global scope, eventually restricted to a class or set of classes by range or domain specifications.

RDF Schema properties are used to denote attributes with constraints about what kind of values a property has. They define an element as an instance of a class by the property `rdf:type` indicating that the element is "of the type" the class define, and stating that a class is a subclass of another by the property `rdfs:subClassOf`. Some of the other properties available to create vocabularies are shown in Table 2.4. New classes can be defined stating that they are instances of the class `rdfs:Class` or subclass(`rdfs:subClassOf`) of another class that is already defined. New properties are defining by stating that they are instances of the class `rdf:Property` or `rdfs:subPropertyOf` another property already defined.

Figure 2.5 depicts an RDF Schema with classes `Recipe`, `Ingredient`, and `Beef`. The `Recipe4600` was defined as an instance of `Recipe` and ingredient `beef` was defined as an instance of `Beef`. The properties `title` and `hasIngredient` denote attributes. The property `hasIngredient` was created to indicate the relation between a recipe and its ingredient. `rdfs:range` and `rdfs:domain` restrictions were specified, stating that only instances of the class `Ingredient` can be related to instances of `Recipe` by this property.

Figure 2.6 gives the RDF/XML format to express the RDF Schema in Figure 2.5. Each RDF markup document starts with the declaration <`?xml...`>. The `rdf:RDF` element is a simple wrapper that marks the boundaries in an XML document between which the content is explicitly intended to mappable into an RDF data model instance. Each RDF markup must be referred to the basic resources of the RDF model and syntax with `xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#."`

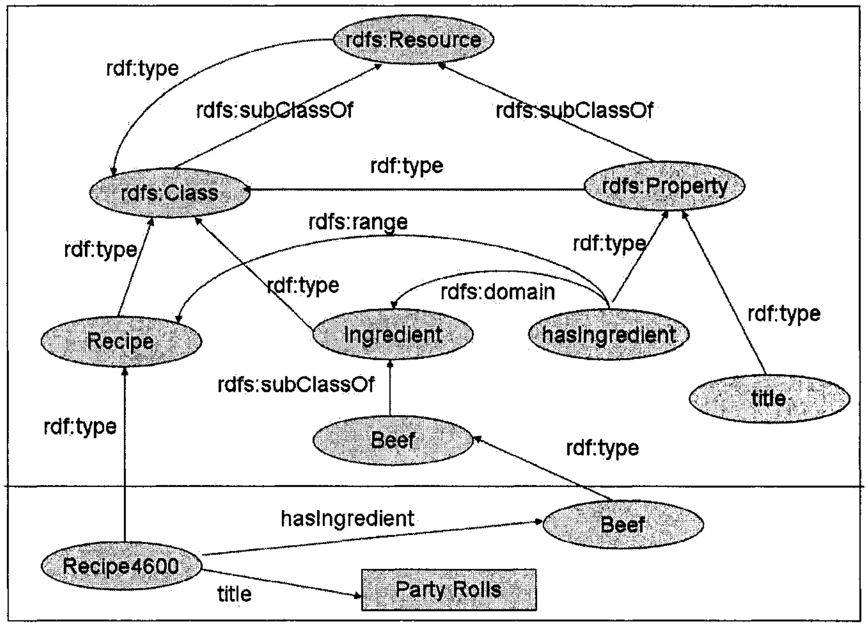


Figure 2.5: Classes and instances in a RDF Schema ontology

This namespace defines the RDF resources, resource type, property and its value. Other namespaces can clarify and identify the data repository. In a RDF document, concepts and relation type names must be prefixed with a unique URI to prevent name clashes from different schemas. The particular RDF document may be checked against concepts defined in RDF Schema to determine consistency. In our application, we use default namespace `xmlns:="http://localhost/cogito/schema/#recipe"` to describe the properties of the specific resource "recipe image." The namespace expresses the concepts belonging to the recipe domain, not others, so it can be understood by a remote machine. The `<rdf:Description></rdf:Description>` tags enclose concepts and properties described by the RDF Schema. Everything within these tags is intended to be interpreted as a RDF data model instance. The identifier of the resource is determined by the `about` attribute. Instead of the `about` attribute, a `Description` tag can have an `ID` attribute if the resource does not have a URI of its own. For clarification, `rdf:ID="Beef"` is equivalent to `rdf:about="#Beef."` It is also possible to represent instances in a compact way.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3c.org/2000/01/rdf-schema#"
         xmlns:dc="http://purl.org/dc/elements/1.1#"
         xmlns:recipe="http://localhost/cogito/schema/#recipe">
<-- Classes-->
<rdf:Description rdf:ID="Recipe">
     <rdf:type rdf:resource="&rdfs;Class"/>
     <rdfs:label>Recipe</rdfs:label>
     <rdfs:comment>used to define web recipes</rdfs:comment>
</rdf:Description>
<rdf:Description rdf:ID="Ingredient">
     <rdf:type rdf:resource="&rdfs;Class"/>
     <rdfs:label>Ingredient</rdfs:label>
     <rdfs:comment>main ingredient in the recipe</rdfs:comment>
</rdf:Desciption>
<rdf:Description rdf:ID="Beef">
     <rdf:type rdf:resource="&rdfs;Class"/>
     <rdfs:label>Beef</rdfs:label>
     <rdfs:comment>a kind of ingredient of the recipe</rdfs:comment>
</rdf:Description>
<--Properties-->
<rdf:Description rdf:ID="hasIngredient">
     <rdf:type rdf:resource="&rdfs;Property"/>
     <rdfs:domain rdf:resource="#Ingredient"/>
     <rdfs:range rdf:resource="#Recipe"/>
     <rdfs:label>has Ingredient</rdfs:label>
</rdf:Desciption>
<!-- an instance of recipe -->
<Recipe about=
         "http://www.cooking.com/recipes/static/recipe4600.htm">
     <dc:title>Party Rolls</dc:title>
     <rdf:type rdf:resource=
         "http://www.w3.org/2000/01/rdf-schema#Resource"/>
     <hasIngredient rdf:resource="#Beef"/>
</Recipe>
</rdf:RDF>
```

Figure 2.6: RDF/XML document to express RDF Schema in Figure 2.5.

| id | Property | Resource | Value |
|---|---|---|---|
| 1 | title | `http://www.cooking.com/recipes/static/recipe4600.htm` | Party Rolls |
| 2 | course | `http://www.cooking.com/recipes/static/recipe4600.htm` | Appetizer |
| 3 | ingredient | `http://www.cooking.com/recipes/static/recipe4600.htm` | Beef |
| 4 | method | `http://www.cooking.com/recipes/static/recipe4600.htm` | Fry |
| 5 | ingredient | `http://www.cooking.com/recipes/static/recipe4600.htm` | Onion |
| 6 | title | `http://food.epicurious.com/run/recipe/view?id2713` | Apple pork chops |

Table 2.5: The semantic data storing in triple store.

RDF/RDF Schema are adequate to model metadata about web resources. By associating a property with a resource at any time, RDF enables the evolution of the metadata description structure.

### 2.1.4 Storage and Retrieval

Metadata describes the content of an application with focus on information that is needed for further usage. It is commonly stored in a database system for quick retrieval and linked to the objects they describe. There are two ways to store the metadata on a computer: an XML-based document or a relational database.

XML-based languages (e.g., XFML, RDF/RDFS) are standard formats for storing metadata in a machine-readable forms (e.g., see Figure 2.4). They are plain text and can be created and edited by hand without any special tools. They are easy to store, transmit, and manipulate. Any of the XML libraries can be used to handle them.

The other option is to store the metadata in a relational database [7]. The relational database relies on the assumption that the data is table-oriented and adheres to a schema that has been defined in advance. There are many SQL libraries available for all major programming languages. The design of the database is based on an Entity-Relationship model [42]. For example, Denton [16] describes how to design the database for storing faceted metadata. RDF triple store (see Section 2.1.3), as another example, also uses an E-R model to design the database schema and store semantic data in a table with Resource, Property, and Value as attributes. Table 2.5 shows a sample of data stored in triple store.

The triple store is powerful for storing data. As shown in Table  2.5, each recipe

```
SELECT          t1.resource
FROM            triple t1, triple t2
WHERE           t1.value = "Appetizer" AND
                t1.property = "course" AND
                t2.value = "beef" AND
                t2.property = "ingredient" AND
                t1.resource = t2.resource
```

Table 2.6: SQL syntax to query a triple store.

is identified by a specific resource with a URI. It is simple to add another property `method` into the database without changing others (see line id=4). For the multiple ingredients in a recipe, it is easy to add to the triple store (see line id=5) without an additional table. However, queries of this triple store are not simple and intuitive, especially for reified RDF data. The queries need self-joins for each field we want to constrain. For example, Table 2.6 gives a query for finding recipes with ingredient `beef` and from course `appetizer`. However, if the multi-properties are used to constrain the queries, the cost of the self-joins must reduce the flexibility of the triple store.

The retrieval process of metadata is to retrieve expected information. Metadata stored in databases contains the searchable information. The focus of the searchable metadata is data useful for fast and easy data matching. The query criteria specified by the user is sent to the databases. The matching algorithms compare the query specification with data stored in databases and retrieves equal or similar results to the user. If database contains records with same data as specified in the query, it simply returns equal records. For example, if the user searches for all recipes with ingredient `beef`, the matching algorithm just has to return all records which contain exactly ingredient `beef`. However, in most cases the exact matching does not work because database may not contain any records that exactly match the query specification, especially on complex search data; or the user just wants to find similar records for further browsing. In this case, the main task involves similarity matching to rank the search results.

Similarity analysis is a primary concern in multimedia database systems where users are interested in retrieving objects which best match the query conditions. The matching algorithms will employ the query metadata for computing the degrees of relevance, and presenting a set of relevant (ranked) results.

Much work has been done on similarity measures. Rahm et al. [37] provides the most recent survey on matching solutions, and describes some of this work in detail. For example, the SemInt system [31] uses a neural network to identify similar attributes from different schemas, matching schema elements based on field specifications (e.g., data types, scale, the existence of constraints) and statistics of data content (e.g., maximum, minimum, average and variance). The Automatch systems [5] use a Naive Bayes learning approach that exploits data instances to match elements. The CAIMAN system [29] computes the similarity between two taxonomic nodes based on their signature TF/IDF vectors, which are computed from the data instances.

Information Retrieval [39] research also provides a set of concepts, models, and methods that are useful for computing semantic similarity. For example, the vector space model [39] is one of the most popular models in Information Retrieval. Its popularity comes from the fact that both document collection and queries are mapped to high-dimension vectors of weighted terms. The similarity can be obtained by computing the distance between the user's query and documents in the collection. For example, a document can be represented as a vector $d = w_{od}, w_{1d}, \ldots, w_{n-1,d}$, and the query can be represented as $q = w_{0q}, w_{1q}, \ldots, w_{n-1,q}$. The similarity between the document and the query can be computed as follows:

$$Sim(q, d_j) = \frac{\sum_{j=1}^{n}(w_{ij} * w_{iq})}{\sqrt{\sum_{i=1}^{n}(w_{ij})^2 * \sum_{i=1}^{n}(w_{iq})^2}} \tag{2.1}$$

The weight $w_{ij}$ could be a simple binary coordinate, i.e., 1 if term $i$ appears in document $j$, or 0 otherwise. It could be adjusted following some specific weighting scheme. A common approach is to consider the product:

$$w_{ij} = tf_{ij} * idf_i,$$

where $tf_{ij}$ - term frequency of term $i$ in document $j$, and

$idf_i$ - inverse document frequency showing how rare is term $i$ in the entire collection of documents.

The inverse document frequency can be computed as:

$$idf_i = \log_2 \frac{N}{n_i},$$

where $N$ - number of documents in the collection, and

$n_i$ - number of documents containing term $i$.

The similarity is produced to infer how much resemblance exists between the user's query and documents in the collection. The returned result of a query is a list which is sorted by the values of the similarity with respect to the query object, even if they match the query only partially. We can specify a threshold to limit the size of the returned result set and sort direction. The more concepts the object shares with the query, the better match it is.

## 2.2 Approaches for Improving Web Search

Web search engines provide search facilities for accessing web sites. However, they present several problems regarding semantic ambiguity. For example, it is hard to know whether a recipe has beef as a significant ingredient and serves as an appetizer course. Currently, many approaches [47, 49] address this problem of locating relevant information and helping the users to explore and understand their search results.

### 2.2.1 Non-Metadata

Non-metadata approaches to organize search results include relevant ranking and clustering [35]. These techniques represent each document as a vector of all words that appear in the document.

Relevant ranking systems [50, 39] create a ranked list of pages in response to a user's search query. Even if the documents are ranked by relevance criteria, an ordered list does not give the user much information on the similarities or differences in the contents of the documents.

Clustering [47] is currently one of most crucial techniques to group the search results automatically in an attempt to provide guidance on how next to proceed. Based on the associations among the documents, clustering separates unrelated pages and clusters related pages into semantic groups which are useful for organization and navigation of web pages. These groups are then presented with the original document references. The user can simply look at the set of groups and get insight into the subjects the query covered. Various clustering algorithms, such as k-means and agglomerative hierarchical clustering [23], are used to determine the degree of association among documents.

Although the clustering approach drew considerable interest from researchers, the results of clustering are difficult to interpret and of limited value in site searching [44]. The clustering groups may not correspond well to the user's query because of its unsupervised learning mechanism. Clustering algorithms usually do not use information about the user's query in forming the clustering groups.

### 2.2.2 Metadata

Metadata has primarily been used as a means to improve web search. It generally uses a controlled vocabulary and provides a scope for locating useful information with the best recall and precision [2]. Much research has looked closely at coupling search results with structured metadata to improve the presentation of search results [6, 46, 49]. Metadata offers a web-wide lookup facility associated with each retrieval page which allows the user to browse directly, or be used to help organize search results. Specifically, faceted metadata provides multiple dimensions to view and navigate the information [49]. The user can see exactly the options available at any time and switch easily between searching and browsing.

## 2.3 Metadata-Based Retrieval Systems

Much work deals with the use of metadata to improve searches from scratch [28, 32, 41, 44, 49]. These studies share the same ultimate goal of utilizing metadata. These systems allow the user to express query data using metadata. However, each system tackles different issues pertaining to metadata, such as using metadata to enrich the search results, to represent and view a resource in multiple dimensions, or to connect and visualize information. This section discusses three such projects.

### 2.3.1 Haystack

Haystack [28] is a system designed to improve the way people manage all the information they work with on daily basis. Haystack uses an RDF data model to incorporate whatever new attributes or relationships are important to the user through a single uniform interface. It deals with annotation and construction of metadata from scratch, allowing the user to organize data closer to her needs, and make comments and annotations for any file.

Based on Semantic Web, Haystack allows the user to use metadata to connect and visualize any kind of information on the web. This might be the name of the author, article she wrote and her email, or any other kind of information. It focuses on information itself, not the programs with which it is usually associated. So only one application should be enough to see both an article and the email of the author who wrote it. Thus, a user could build her own links to semantic objects, which could then be viewed as web pages, taxonomies, etc. However, Haystack needs the user to have some background knowledge to do annotations and searches. It is not appropriate for untrained users.

### 2.3.2 Promootori

Promootori [25] is a semantic image retrieval and browsing system which provides a museum guest with the means to find photographs related to the historical promotion ceremonies of the University of Helsinki. It also allows the users to easily digitize and organize their photos into categories of events based on metadata created by the user, such as the date and time that the photos were taken. The metadata gives the user an overview of the whole ceremony process and the vocabulary for formulating queries. Promootori uses the Semantic Web approach RDF(S) to provide a semantic description for image collections and to enrich the instance-level metadata semantically. It combines the benefits of facet-based and ontology-based search methods, providing a facet-based interface by which the user can easily get an overview of the database contents, learn categories in use, and formulate the queries.

The Promootori interface provides the user with two services. Firstly, the RDF(S) ontologies are projected into facets that facilitate view-based information retrieval. The views provide the user with an overview of the repository contents and a vocabulary for expressing search queries. Secondly, after finding an image of interest by a multi-facet search, ontologies and annotation data are used to recommend that the user view other related data resources, shown as hypertext links. The labels of the links are used to explain the semantic relation to the user.

The idea of facet-based annotation and search has been applied in Promootori. Promootori uses RDF(S) to provide a semantic description of the collection. Promootori requires the user have some Semantic Web knowledge to do annotation and search. Again, it is not appropriate for untrained users.

### 2.3.3 Flamenco

Flamenco(FLexible Access to MEtadata in NOvel Combinations) [19] is a prototype system developed by Elliott at the University of California, Berkeley, which allows users to access a large online image collection based on metadata. The Flamenco project investigates how to effectively incorporate a finite set of faceted metadata into information (including location, architect, style, kind of building, etc.) access user interfaces. Information is gathered based on this metadata. The system uses this metadata for filtering search results together with suggestions of alternative terms to refine queries.

Instead of imposing any special query syntax on the user, Flamenco allows users to navigate through the collections by performing point-and-click interactions. Each selection of a category from a facet narrows the result set, imposing an implicit AND across facets. Flamenco can dynamically generate query previews by explicit exposure of faceted metadata. At each stage in the process, the user sees a preview of the number of results that are assigned to each of the remaining (not yet selected) facets, along with a list of the titles of matching results. Thus, users likely do not feel lost when they are given metadata feedback in the current query state. The interface guides the users toward possible choices by refining and expanding the current query, while maintaining a consistent representation of the collection's structure. However, Flamenco does not group the result set according to the metadata previews. The users cannot evaluate the samples right away until they make final selections.

Siderean[44] provides the same function as Flamenco. It uses the multidimensional attributes of faceted metadata to chart the content of a site as a user zooms in on what he/she is looking for, and as an answer to this, provides a Semantic Web approach to annotate the faceted metadata.

## 2.4 Exploratory Interaction (*Cogito*)

As designed by Hepting [24], *cogito* is a user-centered interactive interface for supporting both user involvement and exploration with the means to assist a user in making an informed choice from a potentially large space of available alternatives. It was implemented in C++.

*Cogito* uses a generic description, expressed as component/element pairs, to describe the choices available within an application of interest. Here, component refers to a logically distinct unit in an object, and element refers to the qualitatively different choices within each component. Each component is instantiated by one or more elements. The Cartesian product of the elements from all components forms the N-dimensional space of possible objects [24].

Figure 2.7 shows a schematic of the *cogito* interface, with a subset of available representations from the current space with current organizational view. Each view consists of one or more screens, and each screen has one or more cells that display representations. The interface displays the objects by sequentially sampling the selected component's elements and finding a representative of each. A user may choose to create several different views of the same space by choosing different views of key components of organization. The user can interact with these candidate representations by clicking directly on the desired cell to iteratively select and evaluate them

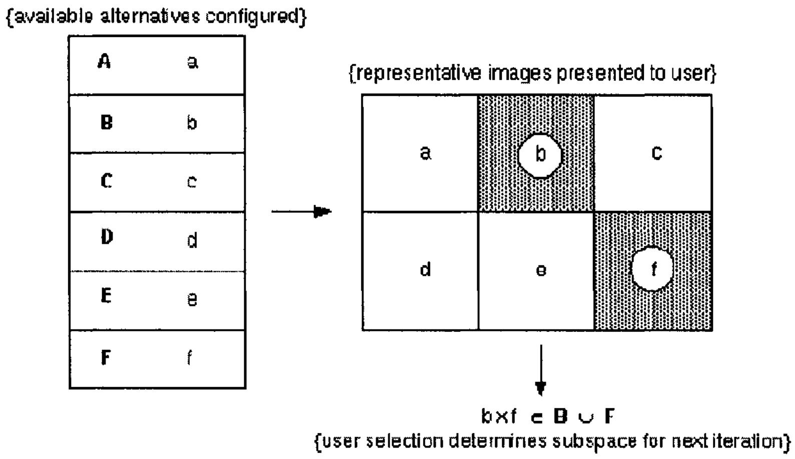

{available alternatives configured}
A a
B b
C c
D d
E e
F f
{representative images presented to user}
a
b
c
d
e
f
bxf ⊂ B ∪ F
{user selection determines subspace for next iteration}

Figure 2.7: Schematic look at the interface: the space of available alternatives is grouped according to user-specified criteria. Each group (A—F) has a representative element (a—f) which is displayed to the user. The subspace for the next search iteration is based on the user selection (b and f)[24].

until promising alternatives can be identified. This process is driven by a genetic approach in which the crossover operation is applied amongst selected combinations. The space of available representations can be very large and it can be difficult to find a desired object. The user can narrow down the space by choosing to limit elements from each component by making selections with respect to personal criteria. The system will generate new alternatives, consistent with which selections are made. Therefore, the conceptual space of possible objects can be effectively navigated under the user's control[24].

## 2.5 Summary

Metadata contains relevant data and structure that is flexible enough to be adaptable for different areas of application. It can support the organization, management, and use of information resources. Due to the simplicity, flexibility, interoperability, and standardization of the metadata, it is easy to properly create it and enables the user to work with the same standard to identify, select, and find desired objects.

Ontology-based languages and standards for metadata structure enable interoperability at the semantic level.

Metadata can be a means to provide effective information discovery in the current web search environment. In contrast to hierarchical metadata, faceted metadata represents and views a resource in multiple dimensions. Recent research has shown that faceted metadata can markedly improve web site searches, especially for large collections of similar-style items that are subject to continuous expansion and change [16, 6, 46, 49]. Not only does faceted metadata provide a standard vocabulary for information annotation and retrieval, it allows the user to dynamically navigate information. Ordinary users have been shown to prefer the faceted metadata approach over clustering [35, 44, 49].

# Chapter 3

# Design and Implementation

The problems described in Chapter 1 are to be solved using recipe metadata. The solution comprises the following aspects:

- Main task: Annotation and retrieval web-based information
- Domain: Recipe
- Original data: Recipes on the Web
- Language for semantic metadata description: XFML
- Programming language: Java
- Database: MySQL Server

This chapter describes the design and implementation aspects of the project and discusses the main modules in detail.

## 3.1 System Architecture

Figure 3.1 shows the architecture of the prototype system. It is designed based on a three-tier client-server architecture [1], where an application is divided into three layers: *The presentation tier (or user interface)* provides easy annotation and retrieval services for the user; *The data tier* consists of the database which is a repository that stores relevant data and makes the annotated data searchable and manageable. Here database refers to a MySQL server. The middle tier (or *core*) provides the process of user services. It is responsible for handling interaction amongst the facet space, the user interface, and the database. It keeps the complexity of the system away from the user.

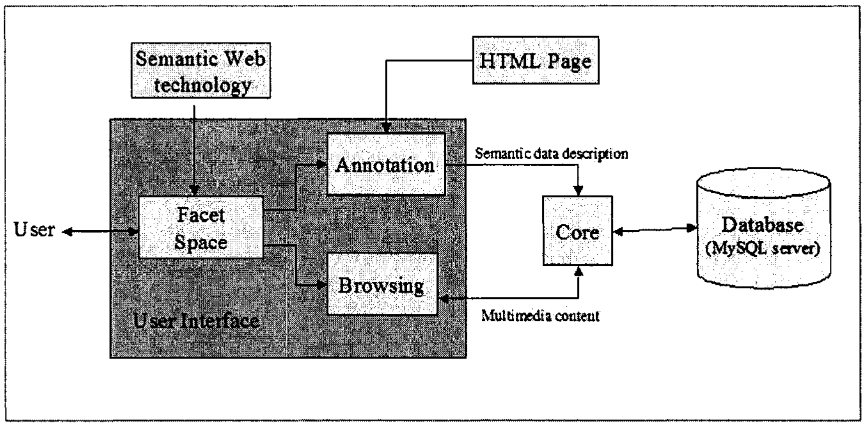
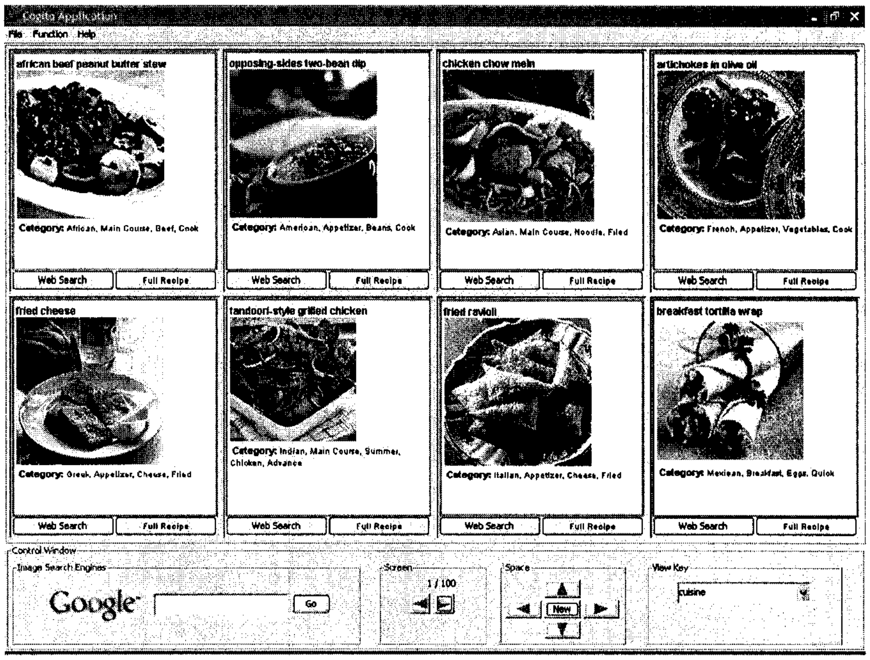


Figure 3.1: The architecture of the prototype system

The application is represented by the *facet space*. During the process of semantic annotation, the user can refer to the facet space to understand the structure of the domain and create meaningful annotations of web-based content. The role and identification mechanism of the user is needed to ensure the storage of high quality information. Each time that a new recipe is annotated, the core module will generate recipe metadata as instances of the facet space with reference to the original documents, and store them in the database. In the retrieval process, the user can interact through the user interface to browse the content of objects, specify queries, and retrieve the recipe objects of interest.

## 3.2 GUI Description

As the front end of the system, the user interface extends the basic *cogito* system [24] to handle a web-based application. It is designed to provide easy annotation and retrieval facilities. Figure 3.2 shows the GUI of the system for the recipe application.



african beef peanut butter stew
Category: African, Main Course, Beef, Cook

opposing-sides two-bean dip
Category: American, Appetizer, Beans, Cook

chicken chow mein
Category: Asian, Main Course, Noodle, Fried

artichokes in olive oil
Category: French, Appetizer, Vegetables, Cook

fried cheese
Category: Greek, Appetizer, Cheese, Fried

tandoori-style grilled chicken
Category: Indian, Main Course, Summer, Chicken, Advance

fried ravioli
Category: Italian, Appetizer, Cheese, Fried

breakfast tortilla wrap
Category: Mexican, Breakfast, Eggs, Quick

Figure 3.2: The GUI for the recipe application.

- Main menu: enables operations such as loading recipe data, connecting to the MySQL database, displaying information, and exiting system operation.

- Screen panel: presents a set of recipes which are organized in sequence by the view key facet. Figure 3.2 shows the current view key facet `cuisine` that is divided ethnically or regionally (`African, American, Asian, French, Greek, Indian, Italian, Mexican, Spanish`). The result set of recipes is organized in sequence by the `cuisine` facet. The user will see a representative sample recipe object from each facet value in turn. The values from the other facets are chosen in a random way.

  - *Each cell* displays a recipe with facet cues and associated finished-dish images, with which the user can interact by clicking directly on it to evaluate

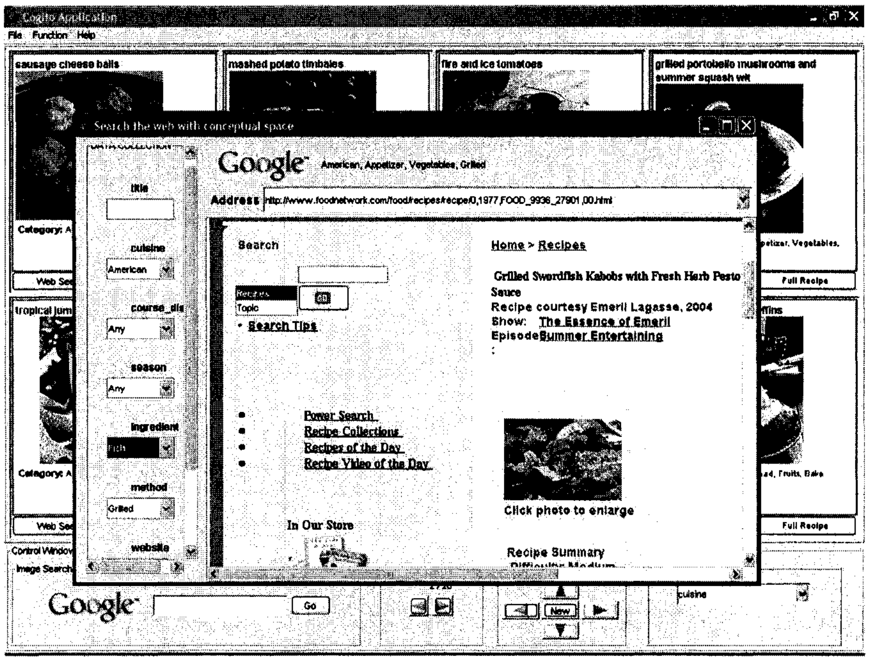


Figure 3.3: The GUI with web search functions for web-based recipe search and annotation.

the recipe and make a selection. If the object is selected, the border of its cell will be turned green to denote its selected state.

- *The Web search button* shows web recipes returned from an embedded web search query made with the selected metadata as search cues (see Figure 3.3). The user can use the integrated annotation function to annotate the content of the web recipes, and add them to the database for later searches.

- *The full recipe button* shows the content of the selected recipe. The recipe reference (URL) refers to original recipe site, so that the user can check the whole recipe for more detail, including current reviews. The Google API tool [21] was used to integrate the web search functionality into the interface.

- Control panel: enables the user to browse the recipe collection.

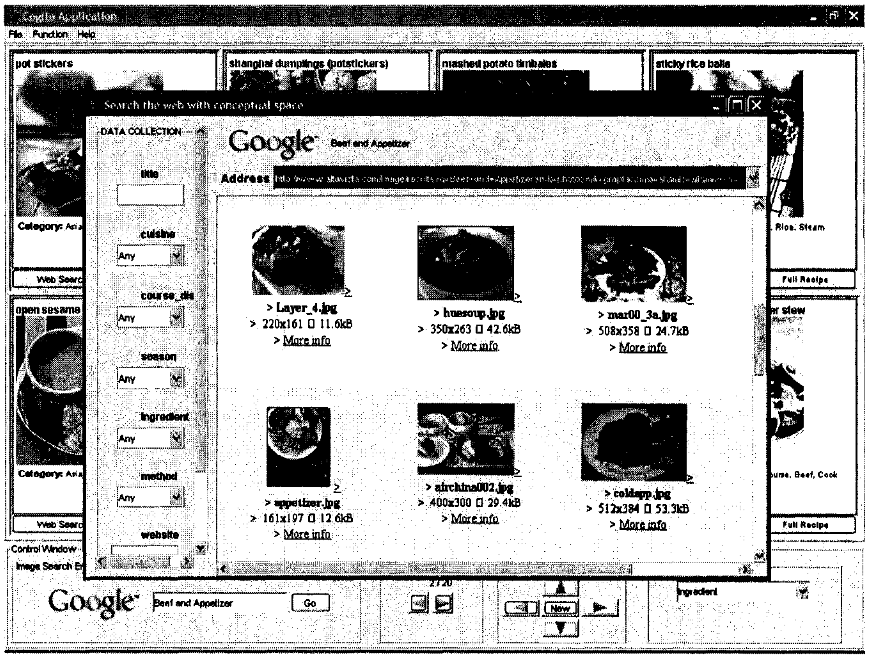


Figure 3.4: The GUI with image web search functions for web-based recipe search and annotation

- *Keyword-based search* allows the user to enter any keyword to locate information from the web and shows results in a list (see Figure 3.4).
- *Screen navigation* allows the user to navigate the current space by selecting different screens.
- *Space navigation* allows the user not only to navigate the history of the space navigation, but also provides a means to directly specify a new query in terms of the defined facets and values.
- *View key* allows the user to view the collection according to different organization schemes.

## 3.3 Data Exchange Format (XFML)

XFML, an XML-based language, is the data exchange format used in this work. XFML is used because it is a very specific and focused format that only expresses the metadata in the form of orthogonal facets. It is easy to implement even using a text editor. XFML allows the user to reuse indexing efforts, which means the user does not have to index the entire web, she can reuse parts of other XFML maps. It is also used to generate and parse the query space that is sent to the database.

In this project, the JDOM [11] technology was chosen to work with XFML. Its implementation enables the construction of the appropriate data structures while parsing the document, and enables the change of the XML tree in the Java environment by adding or deleting nodes.

## 3.4 Annotation

We have given a brief introduction for faceted metadata in Chapter 2. Generally speaking, a domain can be interpreted as a finite set of orthogonal facets. Each facet describes a distinct aspect of the domain and consists of a terminology, e.g., a finite set of names or terms. Below we give a formal definition for a facet space.

**Definition 3.1.** *A facet space for a specific domain is defined by a finite orthogonal set of facets $\mathcal{F} = \{F_1, F_2, \ldots, F_n\}$, where each facet $F_i = \{V_1, V_2, \ldots, V_m\}$ consists of a taxonomy*[1] *over the facet, such as a finite set of terms or names, n is the number of the facets which constitute the N-dimensional concept space.*

The facet space expresses all semantic information explicitly to be accessible and processable by our prototype system. It is encoded in the formal ontology-based language XFML.

For the recipe application, the process of constructing the facet space can be started by selecting orthogonal facets, like `course, cuisine, ingredient, method, season,` and `special` to describe the content of recipes. Each facet consists of a finite set of values, describing a recipe from a particular aspect. For example, the facet `cuisine` has values of: `African, American, Asian, French, Greek, Indian, Italian, Mexican,` and `Spanish`. The facet space is encoded in XFML as displayed in Table 3.1. This facet space does not cover the whole area of the application domain.

[1] *A taxonomy is a set of terms that are arranged into a generalization-specialization hierarchy. A taxonomy does not define attributes of these terms, nor does it define relationships between the terms.*

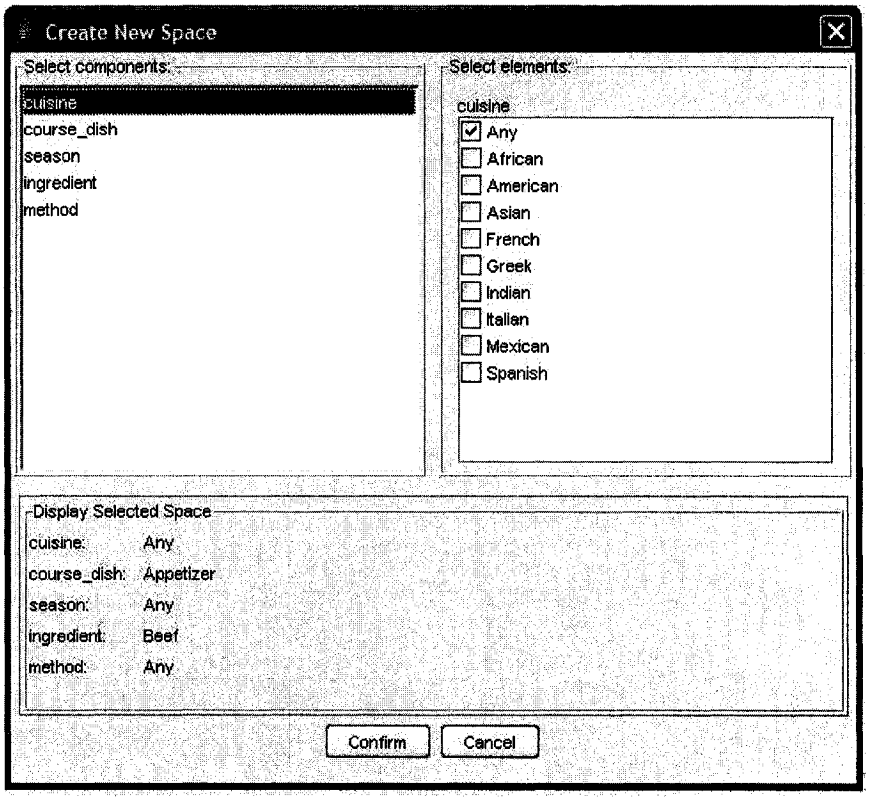
```
<?xml version="1.0"?>
<!DOCTYPE xfml SYSTEM "xfml.dtd">
<xfml version="1.0" url="http://cogito/recipe.xml"
     language="en-us">

<!-- FACETS -->
<facet id="course">course</facet>
<facet id="cuisine">cuisine</facet>
<facet id="ingredient">main ingredient</facet>
<facet id="method">cooking method</facet>
<facet id="season">season</facet>
<facet id="special">special consideration</facet>

<!-- TOPICS -->
<topic id="appetizer" facetid="course">appetizer</topic>
<topic id="beverage" facetid="course">beverage</topic>
<topic id="African" facetid="cuisine">African</topic>
<topic id="Asia" facetid="cuisine">Asia</topic>
<topic id="beef" facetid="ingredient">beef</topic>
<topic id="beans" facetid="ingredient">beans</topic>
<topic id="grilled" facetid="method">grilled</topic>
<topic id="spring" facetid="season">spring</topic>
<topic id="meatless" facetid="special">meatless</topic>
     ...
</xfml>
```

Table 3.1: An annotated facet space for the recipe domain.

It can be enhanced by adding new facets, relations or instances without affecting the whole structure. The built space is shown in Figure 3.4. Such space helps the user understand what she is viewing and enables the user to interact with the user interface to annotate and retrieve recipes.

As discussed in Chapter 2, the Semantic Web is to have machine-understandable metadata on each web page. This metadata is annotated with extremely high quality

.



Select components:

cuisine
course_dish
season
ingredient
method

Select elements:

cuisine

☑ Any
□ African
□ American
□ Asian
□ French
□ Greek
□ Indian
□ Italian
□ Mexican
□ Spanish

Display Selected Space

| | |
|---|---|
| cuisine: | Any |
| course_dish: | Appetizer |
| season: | Any |
| ingredient: | Beef |
| method: | Any |

Figure 3.5: The facet space dialogue panel.

information. The annotation schema built from facet space can help the user understand the content and structure of any given web-based recipe, which has made it feasible to annotate web information intuitively without any domain knowledge. The user interface described in Section 3.2 provides an annotation function for a user to annotate web-based recipes. The data that is used to describe a recipe depends mainly on the data that can be extracted from the Java GUI elements. The Google API tool [21] was used to integrate the web search functionality into the interface. Each time a new recipe is annotated, the core module will generate recipe metadata as instances of the facet space, with reference to the original documents, and store them in the database. Figure 3.6 shows an annotation for a given recipe web site.

To simulate this process, the RecipeCrawler class was implemented to extract a large amount of recipe data from known recipe web sites (such as `cooking.com`

```
<!-- an annotated web-based recipe -->
<page url="http://www.cooking.com/recipes/static/recipe4600.htm">
    <title>Party Rolls</title>
    <creator>Bon Appetit</creator>
    <!  - - use this syntax if the value of the property is concept - ->
    <occurrence topicid="beef"/> <!-- Ingredient facet -->
    <occurrence topicid="appetizer"/> <!-- Course facet -->
    <occurrence topicid="summer"/> <!-- Season facet -->
    <occurrence topicid="fry"/> <!-- cooking method -->
    <occurrence topicid="kid-friendly"/> <!-- special -->
    <occurrence topicid="American"/> <!-- cuisine -->
</page>
```

Figure 3.6: An annotation for a given recipe web site.

and epicurious.com). These sites show a set of recipes. Each of them describes a recipe which is in our facet space. The recipecrawler attempts to automatically separate that web page into distinct recipe records by identifying the boundary tags such as <table> and <tr>, and strips out all markup in each record, leaving only plain text content. The values of facets for a recipe descriptions can be generated by following the category classification path for each recipe. An address (URL) is used as a pointer to the extracted data, preserving the link between the extracted data and the source document. Due to the fact that most recipe web sites use different formats, this crawler had to be tailored in very specific ways to adapt to other applications. Figure 3.7 shows an algorithm used in the RecipeCrawler class.

Once the system finishes extracting the data, storing it in a database is straightforward. The database generated by the system has a main table, where each row represents a recipe in the facet space. It is a repository that not only stores the HTML pages of recipe content with their references (URLs), but also contains semantic descriptions of recipes. It makes the annotated data searchable and manageable. The choice of only storing references to recipe objects makes the system agile and responsive to changes on the web. For example, if a recipe from epicurious.com is unrated

```
Given:
    a web site URL with a set of links (seed)
    a facet space
Algorithm:
    seed _ url := getUrl(GoogleAPI);
    concepts := getSpace(Document );
    url _ links := fetchWebpageLink(seed _ url)
    while(#url(url _ links) <> null){
         page := processWebpageLink(url){
             links := fetchWebpageLink(page);
             for each link {
                  if link contains "recipe"{
                     recipe := processRecipe(link, concepts);
                     add recipe to the collection ;
                  }else{
                     Go to processWebpageLink(link);
                  }
             }
         }
    }
    return collection;
```

Figure 3.7: A pseudo-code for collecting web-based recipe data.

when first viewed, it might be bypassed. However, on a subsequent search the user sees several very high ratings for a recipe, she might be inclined to try it. Reference is like a browser's bookmark facility.

## 3.5 Retrieval Process

The semantic data from an annotation process is stored in the searchable database. Our system has to enable an easy and intuitive query specification for the user, retrieve certain results from the database according to matching algorithm, preprocess the raw results (i.e., rank, delete duplicates, and organize), and display them for further use.

### 3.5.1 Query Specification

The facet space gives the user an overview of the kind of information that is in the database and guides the user in formulating the query in terms of appropriate facets. Furthermore, facets provide multiple dimensions in which to browse the database content. The following shows how a query can be specified by the user through the user interface.

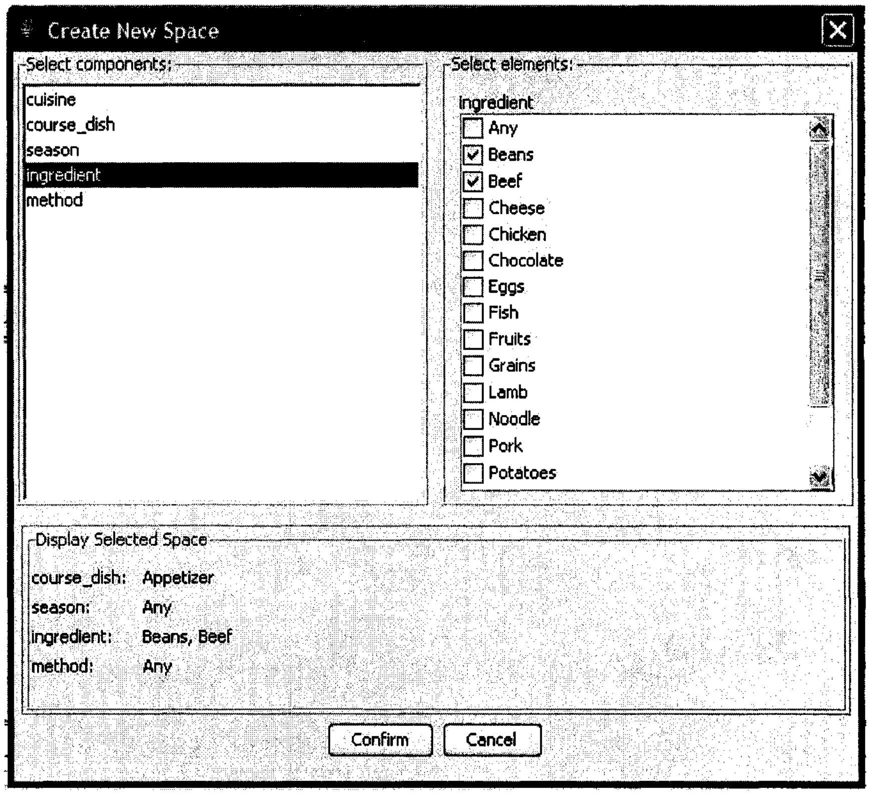
- Select promising recipes from the screen panel to form a query space.
- Edit query space of interest by opening the edit panel of facet space.
- Submit the query by pressing the confirm button.

The user's query is sent to a query generation module which automatically generates a query data without user interaction. In our work, the query is constructed in XML-based format. For example, Figure 3.8 shows a user's query: "Show me all recipes that have ingredient `Beans` or `Beef` serving for appetizer course, sorted by cuisine." This query forms two dimensional space of $2 \times 1$ available representations. The query can be expressed in XML-based format as follows:

```
<query sortby="cuisine">
   <facet id="course">course</facet>
   <facet id="ingredient">main ingredient</facet>
   <topic id="beef" facetid="ingredient">Beef</topic>
   <topic id="beans" facetid="ingredient">Beans</topic>
   <topic id="appetizer" facetid="course">Appetizer</topic>
</query>
```

### 3.5.2 Space Matching

To answer the user's query, the system retrieves certain results from the database according to the matching algorithm.



Ingredient

- ☐ Any
- ☑ Beans
- ☑ Beef
- ☐ Cheese
- ☐ Chicken
- ☐ Chocolate
- ☐ Eggs
- ☐ Fish
- ☐ Fruits
- ☐ Grains
- ☐ Lamb
- ☐ Noodle
- ☐ Pork
- ☐ Potatoes

Display Selected Space

```
course_dish:  Appetizer
season:       Any
ingredient:   Beans, Beef
method:       Any
```

Figure 3.8: A query space generated from the facet space panel

The query space is usually mapped to certain fields according to a specific database specification imposing an implicit *AND* across facets. If the facets selected are $F_1, F_2, \ldots, F_n$, and the values of $F_i$ are $T_{i,1}, T_{i,2}, \ldots, T_{i,j}$, $(i = 1, \ldots, n)$, then this query space can be expressed as a disjunctive form with boolean *AND-OR* constraints:

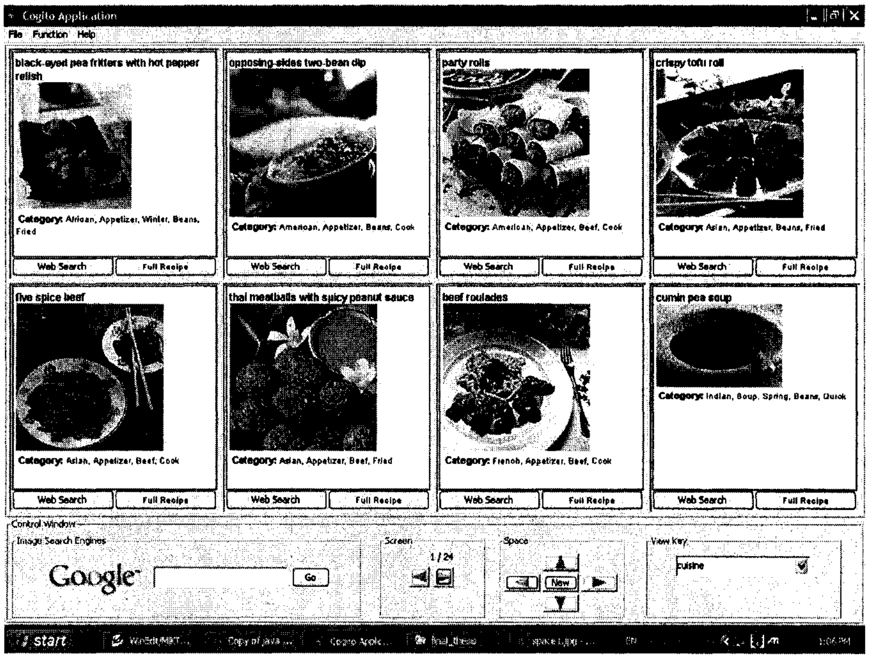
$$(T_{1,1} \bigvee \cdots \bigvee T_{1,j}) \bigwedge (T_{2,1} \bigvee \cdots \bigvee T_{2,j}) \bigwedge \cdots \bigwedge (T_{n,1} \bigvee \cdots \bigvee T_{n,j}),$$

For example, the sample query space can be expressed as a SQL query.

```
SELECT      title
FROM        recipe
WHERE       course="appetizer" AND
            ingredient="beef" OR ingredient="beans"
```

```
ORDER BY    cuisine
```

The returned result is a set of recipes which satisfies the query's criteria: ingredient `beans` and course `appetizer`, ingredient `beef` and course `appetizer`. As shown in Figure 3.5.2, the combination of the space is an approach to help the user to browse unfamiliar concepts and sites, and guide them to search for items of interest. On the other hand, the returned results are sorted by a facet and using `ORDER BY` operators.



black-eyed pea fritters with hot pepper relish
Category: African, Appetizer, Winter, Beans, Fried

opposing-sides two-bean dip
Category: American, Appetizer, Beans, Cook

party rolls
Category: American, Appetizer, Beef, Cook

crispy tofu roll
Category: Asian, Appetizer, Beans, Fried

five spice beef
Category: Asian, Appetizer, Beef, Cook

thai meatballs with spicy peanut sauce
Category: Asian, Appetizer, Beef, Fried

beef roulades
Category: French, Appetizer, Beef, Cook

cumin pea soup
Category: Indian, Soup, Spring, Beans, Quick

View Key: cuisine

Figure 3.9: The GUI with results from the query space "ingredient (beef, beans) and course(appetizer)"

However, the exact match to answer the user's query is inadequate because identical recipes might be described in different ways. To avoid negative consequences like empty results or a smaller result set, the similarity-based retrieval is introduced as opposed to the exact match search. The similarity can be obtained by computing the distance between the user's query and semantic recipe descriptions in the collection. As discussed in Section 2.1.4, there are various methods for performing similarity matching, e.g., Cosine distance.

Since the dimension of the data collection may be large, and we need to ensure fast query processing, we do not attempt to compare all members of the collection against one another (as required by Equation 2.1). Rather, we produce a similarity score by facet matching, as in Equation 3.2. In our system, the query space is generated automatically while the user browses, but for now we assume all facets are assigned equal weight:

$$f = \frac{\sum i f_i}{N}, f \in (0,1), \tag{3.2}$$

where N - the total number of facets in the query.

For each facet

$$f_i = \begin{cases} 1 & \text{if object contains the same category as query} \\ 0 & \text{otherwise.} \end{cases}$$

The scoring scheme $f_i$ for each of the facets is produced to infer how much resemblance these is between the pair of query and each record of data collection. The returned results of a query are a list sorted by similarity with respect to the query object, and guide the user search further. We can specify a threshold to limit the size of the returned result set and sort direction. The more concepts the object shares with the query, the better a match it is. Our implementation will support this function which matches against a certain threshold.

For instance, suppose the user's query is "Find a recipe with a main ingredient of beef, is intended to serve as an Appetizer course, and has a fried cooking method." The query and collection can be represented as a set of vectors as in Figure 3.10a.

From Figure 3.10, we know that recipe d1 contains all concepts in the query, and thus is said to be most similar to the query. But recipe d2 is totally different from the query. It does not contain any concept that appears in the query. It is possible to rank the objects against their scores. The above example shows that objects having a number of common concepts are close to each other. Users can specify a comparison condition. The structure of a conceptual space description has a high impact on the simplicity of similarity score computing. We also can use a Cosine distance function (see Equation 2.1) to compute the similarity, but with the additional costs already noted.

| | Cuisine | Course | Main Ingredient | Season | Cooking Method |
|---|---|---|---|---|---|
| query | | Appetizer | Beef | | Fried |
| d1 | American | Appetizer | Beef | Holiday | Fried |
| d2 | American | Soup | Beans | Summer | Cook |
| d3 | Asia | Main Course | Beef | | Fried |

(a) The vectors of query and recipe objects

| | Course | Main Ingredient | Cooking Method | Score |
|---|---|---|---|---|
| d1 | 1 | 1 | 1 | 1.0 |
| d2 | 0 | 0 | 0 | 0 |
| d3 | 0 | 1 | 1 | 0.67 |

(b) Similarity score

Figure 3.10: The similarity score computed by proposed approach.

## 3.6 Implementation Notes

The system architecture was implemented as proposed. The high-level programming language Java[26] provided by Sun was chosen as the implementation language because it is platform independent and thus the prototype system is not limited to a specific operating system. The user interface was built using Java Swing [13] with component sets (i.e., Button, Scrollbar, Label, TreeView, ListBox, ColourChooser, FileChooser, ProgressBar, etc.). As a back end, MySQL [18] was chosen as the database management system because it is free. It offers excellent storage and Standard Query Language (SQL) query facilities for all major programming languages. The user's queries can be answered directly. MySQL is also a natural choice when one wants to use PHP[48] for web enabled database sites. The connectivity between the relational database MySQL and these application modules was achieved through JDBC [22]. The Google API tool [21] was used to integrate the web search functionality into the interface.

XFML, an XML-based language for exchanging faceted metadata, was used in this work to encode the semantic description and provide a unified exchange base for all modules. In order to parse an XFML file, the JDOM [11] technology was chosen to work with XML, which considers Java specific design patterns.

# Chapter 4

# Evaluation

A usability study was conducted to evaluate how the prototype system works and how well people could use it. In addition to identifying whether users could accurately complete some specified tasks and whether their experience was satisfying, the potential benefit and impact of the prototype system on current practice was also considered.

## 4.1 Experiment Design

In the experiment, the participants were required to do some recipe browsing and searching. The main approach for every experimental session was a pre-task questionnaire, the task itself, and a post-task questionnaire.

### 4.1.1 Data Collection

A recipe collection was chosen for this study because it is easy to understand and is considered to be a good match to the skills expected in the pool of potential participants. To capture relevant information in a tractable way, all participants were given the same facet space (see Table 4.1) describing a recipe domain. No personalization was attempted.

More than 2000 recipes were extracted from unstructured or semi-structured web collections (such as from the web site: `http://www.epicurious.com/`). The extracted semantic data were stored in a database with resource references (URL) for

| Facet | Size | Values |
|---|---|---|
| Cuisine | 9 | African, American, Asia, French, Greek, Indian, Italian, Mexican, Spanish |
| Course | 13 | Appetizer, Beverage, Bread, Breakfast, Brunch, Condiment, Dessert, Hors D'Ouvres, Main course, Salad, Sandwiches, Side dish,Soup |
| Season | 6 | Fall, Holiday, Spring, Summer, Winter |
| Ingredient | 15 | Beans, Beef, Cheese, Chicken, Chocolate, Eggs, Fish, Grains, Lamb, Noodle, Pork, Potatoes, Rice, Shrimp, Vegetables |
| Method | 12 | Advance, Bake, Broiled, Cook, Fried, Grilled, No cook, Quick, Roast, Sauté, Steam, Stir-fry |
| Special | 3 | Kid-friendly, Low fat, Meatless |

Table 4.1: The recipe space used in this study, defined by the Cartesian product of these facets and values, contains 404,352 recipe alternatives. Only a fraction of these might be considered edible.

further usage. The system can handle this amount of recipes.

798 recipes with finished-dish images were chosen from the database for the usability study. The layout of these recipes, along with some results, is shown in Table 4.2.

### 4.1.2 Participants

Twenty participants were recruited through e-mail, from the University of Regina Computer Science Department Participant Pool: undergraduate students who were taking CS classes at the University of Regina. Participants were asked to take about 45 minutes to complete all tasks and questionnaires. In return, each participant received 1 point toward the final grade in one of their participating Computer Science classes. All participants were required to sign a consent form (see Appendix B, page 80) at the start of the session.

| | | Methods | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Advanced | Bake | Broiled | Cook | Fried | Grilled | No-Cook | Quick | Roast | Sauté | Steam | Stir-Fry |
| | | 8 | 43 | 0 | 24 | 18 | 7 | 4 | 14 | 2 | 19 | 0 | 1 |
| Ingredients | | 53 | 204 | 12 | 107 | 55 | 57 | 38 | 90 | 51 | 86 | 21 | 24 |
| Beans | 16 | | | | 1 | | | 1 | 1 | | **12** | | 1 |
| | 42 | 2 | | 1 | 14 | 3 | | 1 | 10 | | **8** | 1 | 2 |
| Beef | 14 | | 3 | | 9 | | | | | 1 | 1 | | |
| | 61 | | 6 | | 11 | 1 | 20 | 1 | 3 | 6 | 10 | 1 | 2 |
| Cheese | 9 | | 5 | | 2 | 1 | | | 1 | | | | |
| | 64 | 3 | 25 | | 5 | 9 | 3 | 9 | 4 | 3 | 3 | | |
| Chicken | 11 | | 6 | | | 2 | 2 | | | | 1 | | |
| | 67 | 3 | 15 | 3 | 9 | 4 | 8 | | 2 | 5 | 13 | 1 | 4 |
| Chocolate | 4 | 1 | 3 | | | | | | | | | | |
| | 26 | 9 | 12 | | | | | 2 | 1 | 1 | | | 1 |
| Eggs | 37 | 5 | 6 | | 6 | **15** | | | 3 | | 2 | | |
| | 76 | 4 | 31 | 1 | 14 | **7** | 1 | 1 | 10 | 1 | 5 | | 1 |
| Fish | 4 | | | | 1 | | 3 | | | | | | |
| | 73 | 3 | 12 | 3 | 9 | 9 | 10 | 1 | 10 | 3 | 7 | 6 | |
| Fruits | 5 | | | | 2 | | | 2 | 1 | | | | |
| | 75 | 8 | 36 | | 3 | 1 | 2 | 9 | 12 | | 4 | | |
| Grains | 2 | | 2 | | | | | | | | | | |
| | 23 | 1 | 10 | 1 | 4 | 2 | | 1 | 1 | | 2 | 1 | |
| Lamb | 1 | | | | | | | | | 1 | | | |
| | 21 | 1 | 3 | | 2 | | 5 | | 2 | 6 | 1 | 1 | |
| Noodles | 0 | | | | | | | | | | | | |
| | 23 | | 3 | | 6 | 3 | | | 4 | 2 | 4 | | 1 |
| Pork | 3 | 2 | 1 | | | | | | | | | | |
| | 39 | 4 | 9 | | 5 | 2 | 2 | | 3 | 7 | 3 | 2 | 2 |
| Potatoes | 2 | | 1 | | | | 1 | | | | | | |
| | 26 | 2 | 7 | | 3 | 2 | 1 | | 5 | 2 | 4 | | |
| Rice | 0 | | | | | | | | | | | | |
| | 21 | 2 | 2 | | 8 | | | | 3 | | 2 | 3 | 1 |
| Shrimp | 3 | | 2 | | | | | | 1 | | | | |
| | 11 | 1 | 2 | | 1 | | 1 | | 1 | | 3 | 1 | 1 |
| Vegetables | 29 | | 14 | | 3 | | 1 | 1 | 7 | | 3 | | |
| | 150 | 10 | 31 | 3 | 13 | 12 | 4 | 13 | 19 | 15 | 17 | 4 | 9 |

Table 4.2: The distribution of recipes within the database, organized based on facets `ingredient` and `method`. Each ingredient/method combination has a pair of numbers: the top number is the number of times that an ingredient/method combination recipe was selected and the bottom number is the total those recipes available. Zero entries are left blank. Entries in bold indicate that more selections were made than the total number of unique available recipes.

### 4.1.3 Pre-Task Questionnaire

The pre-task questionnaire (see Appendix B) was designed to gain background information about the participants (such as gender, age, education, and area of interest).

### 4.1.4 Tasks

A tutorial (see pages 82 – 84 in Appendix B) walked participants through the features of the study interface. The basic functions were explained, and participants were shown how to switch between them. The interface was demonstrated to all participants, and each participant was allowed to practice with it until he or she felt comfortable.

The participants were then asked to complete a set of tasks. The task questions (see Appendix B, page 85) were designed to cover critical functionalities of the user interface based on the faceted metadata description of a typical recipe collection. The participants were asked to search for recipes and to make selections until they were satisfied that the recipes which they selected met the requirements of the particular task question.

During this time, the participants were encouraged to work without guidance, though help was given as needed when the participants asked questions. The experimenter recorded sessions through hand-written notes that included elapsed time for each task, participants' comments, and any problems.

### 4.1.5 Post-task Questionnaire

Once the participants had completed the experiment, they were asked to answer a post-task questionnaire (see Appendix B, pages 86 – 88) which was designed to assess the participants' impressions about the software and the task of finding recipes. The questionnaire comprised 11 subjective rating questions on a four point Likert scale, including ratings of facet importance. Seven open-ended questions were also posed.

## 4.2 Results

The study session data from 20 participants used in the analysis was collected from questionnaires. The Chi-Square ($\chi^2$) test was used to measure the difference between the expected (equal) distribution and the observed distribution in the questionnaire responses.

### 4.2.1 Pre-task

The pre-task questionnaire (see page 81, Appendix B) collected background information from the participants. Table 4.3 shows all responses of the participants.

16 of the 20 participants were from Computer Science, Two participants were from Arts and Two more were from Business. Sixteen of the twenty participants were male. Most participants ranged in age from 18 to 25 years. The participants were all regular users of the Internet, searching for information by utilizing web search engines every day or at least a few times a week. Most of the participants did not search for cooking recipes online too often, with 12 participants searching for recipes less than monthly. The favorite recipe source was fairly equally split between books and the web.

### 4.2.2 Tasks

With each task question, participants were asked to write down the name of their selected recipes. No task questions were assigned time limits. Instead, they were expected to search as they normally would. The time reported is that required to browse and make their selections.

In fact, it is difficult to evaluate browsing tasks because there are no correct answers [49]. Rather, participants were graded on the basis of number of requested facet topics found. The results of the tasks indicated that participants were significantly successful in retrieving relevant recipes and getting correct answers( $p < 0.01$). A complete summary of tasks scores and response times are given in Tables 4.12 and 4.13.

| Topic | Response Category | Results |
|---|---|---|
| Q1. Gender | Male | 16 |
| | Female | 4 |
| Q2. Age | 18-25 | 15 |
| | 26-35 | 4 |
| | 36-49 | 1 |
| | 50+ | 0 |
| Q3. Post-secondary education | 1-4 | 15 |
| | 5-6 | 3 |
| | more than 6 | 2 |
| Q4. Study area | Fine Arts | 0 |
| | Arts | 2 |
| | Business Arts | 2 |
| | (Applied)Science | 16 |
| Q5. Frequency of using computer | Daily | 20 |
| | Weekly | 0 |
| | Monthly | 0 |
| | Less than monthly | 0 |
| Q6. Frequency of using web search engines | Daily | 18 |
| | Weekly | 2 |
| | Monthly | 0 |
| | Less than monthly | 0 |
| Q7. Frequency of searching for cooking recipes online | Daily | 0 |
| | Weekly | 1 |
| | Monthly | 7 |
| | Less than monthly | 12 |
| Q8. Preferred source for cooking recipes | Books | 10 |
| | Web pages | 8 |
| | Magazines/Newspapers | 1 |
| | Other | 1 |

Table 4.3: The Response and Analysis of the pre-task questionnaire based on frequencies of response categories for all participants (N = 20).

| Recipe Name | Score | Times Selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Sesame chicken wings | 1.0 | 3 | Chicken | Bake | American, Appetizer |
| Party rolls | 1.0 | 3 | Beef | Cook | American, Appetizer |
| Christmas lane cake | 1.0 | 2 | Eggs | Bake | American, Dessert, Holiday |
| Scallops on skewers with carrot sauce | 1.0 | 2 | Fish | Grilled | American, Appetizer |
| Parmesan cheese twists | 1.0 | 1 | Cheese | Bake | American, Bread |
| Swiss cheese, spinach and bacon appetizer tarts | 1.0 | 1 | Cheese | Bake | American, Appetizer |
| Tuxedo cheesecake | 1.0 | 1 | Cheese | Bake | American, Dessert |
| Mashed potato timbales | 1.0 | 1 | Potatoes | Bake | American, Appetizer |
| Peanut butter oatmeal cookies | 1.0 | 1 | Eggs | Bake | American, Dessert |
| Mediterranean crescent pinwheels | 1.0 | 1 | Grains | Bake | American, Appetizer |
| Almond sunshine citrus | 1.0 | 1 | Fruits | No cook | American, Appetizer |
| Herbed chicken quarters | 1.0 | 1 | Chicken | Grilled | American, Main Course |
| Marinated beef pot roast | 1.0 | 1 | Beef | Roast | American, Main Course |
| Melanie's garden-tomato soup | 1.0 | 1 | Vegetables | Cook | American, Soup |

Table 4.4: Recipes chosen in Task 1 with American cuisine.

1. **Choose a recipe with American cuisine that you might like to eat.**

   Task 1 was a low-constraint task requiring only a single facet (cuisine = `American`). Participants spent an average of 44 seconds (standard deviation = 26.4) on this task. All participants completed this task successfully. The selected recipes are shown in Table 4.4.

2. **Find a recipe with a fried cook method and a main ingredient of eggs.**

   Task 2 was more difficult because it required matching of 2 facets. Participants spent an average of 51.45 seconds (standard deviation = 24.97) on this task. Only 14 of 20 participants completed this task successfully. The average score, where 1 is a perfect score, was 0.85(standard deviation = 0.24). Table 4.5 shows the selected recipes.

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Fried eggs with vegetable confetti | 1.0 | 6 | Eggs | Fried | American, Brunch, Spring |
| Classic French toast | 1.0 | 6 | Eggs | Fried | French, Breakfast |
| Breakfast stack | 1.0 | 1 | Eggs | Fried | American, Brunch |
| Fried eggs and asparagus with parmesan | 1.0 | 1 | Eggs | Fried | American, Brunch, Spring |
| Tandoori-spiced chicken breasts | 0.5 | 2 | Chicken | Fried | Indian, Main Course |
| Fried Ravioli | 0.5 | 2 | Cheese | Fried | Italian, Appetizer |
| Roasted red pepper and zucchini frittata | 0.5 | 1 | Eggs | Cook | Italian, Main Course |
| Spinach devilled eggs | 0.5 | 1 | Eggs | Quick | American, Appetizer |

Table 4.5: Chosen recipes for Task 2 with a fried cook method and a main ingredient of eggs.

3. **Find a recipe with a main ingredient of beans, that is intended for the summer season, and has a sauté cooking method.**

   Task 3 was a high-constraint task which required participants to match 3 facets. Only 9 of 20 participants received full marks for their answers. The average score was 0.835 (standard deviation = 0.17). Participants took an average of 51.65 seconds (standard deviation = 23.69). The selected recipes are shown in Table 4.6.

4. **Imagine you would like to have eggs for breakfast. Choose a recipe that you might like to eat.**

   Task 4, like Task 2, had two facets as its constraint. Participants took an average of 36.7 seconds (standard deviation = 13.58). 19 of 20 participants completed this task successfully, for an average score of 0.95 (standard deviation = 0.22). Table 4.7 shows the selected recipes.

5. **Imagine you are cooking for a group of people and for the appetizer course, you would like to serve something with French cuisine. Choose one of these recipes that you might like to serve.**

   This task also had 2 facets as its constraint. It took the participants an average

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Succotash | 1.0 | 8 | Beans | Sauté | American,<br>Side Dish,<br>Summer |
| Cavatappi with white beans and golden onions | 1.0 | 1 | Beans | Sauté | American,<br>Main Course,<br>Summer |
| Summer tomato and Basil Spaghettini | 0.67 | 3 | Vegetables | Sauté | Italian,<br>Main Course,<br>Summer |
| Chicken and white bean chili | 0.67 | 1 | Beans | Sauté | American,<br>Main Course,<br>Winter |
| Three way garlic pasta with beans and peppers | 0.67 | 1 | Beans | Sauté | Italian,<br>Main Course,<br>Spring |
| Curried tofu with spinach and tomatoes | 0.67 | 1 | Beans | Sauté | American,<br>Main Course |
| Pork stir-fry with green beans and peanuts | 0.67 | 1 | Beans | Stir-fry | Asian,<br>Main Course,<br>Summer |
| Barbecued ribs with corn and black-eyed-pea salad | 0.67 | 1 | Beans | Quick | American,<br>Main Course,<br>Summer |
| Sautéed skirt steak | 0.67 | 1 | Beef | Sauté | American,<br>Main Course,<br>Summer |
| Spice-rubbed chicken breasts with lemon-shallot sauce | 0.67 | 1 | Chicken | Sauté | American,<br>Main Course,<br>Summer |
| Tuna and white bean salad | 0.33 | 1 | Beans | No Cook | American,<br>Main Course |

Table 4.6: Recipes chosen in Task 3 with ingredient beans, summer season and a sauté cooking method as constraints.

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Eggs florentine | 1.0 | 5 | Eggs | Advance | French, Breakfast |
| Quick eggs benedict | 1.0 | 3 | Eggs | Quick | American, Breakfast |
| Asparagus omelet | 1.0 | 2 | Eggs | Sautè | French, Breakfast |
| Chorizo scrambled eggs | 1.0 | 2 | Eggs | Cook | Spanish,Winter Breakfast |
| Potato pancakes with apple sauce | 1.0 | 2 | Eggs | Cook | American,Breakfast Breakfast |
| Easy breakfast nacho bake | 1.0 | 2 | Eggs | Bake | Mexican, Breakfast |
| Fried eggs on corn tortillas with two salads | 1.0 | 1 | Eggs | Fried | Mexican,Breakfast |
| Breakfast tortilla wrap | 1.0 | 1 | Eggs | Quick | Breakfast |
| Egg stuffed breakfast tomatoes | 1.0 | 1 | Eggs | Bake | French,Breakfast |
| Spinach & mushroom frittata | 0.0 | 1 | Vegetables | Cook | Italian,Brunch |

Table 4.7: Recipes chosen in Task 4 with eggs for breakfast.

of 41.4 seconds (standard deviation = 22.42). All participants completed the task successfully. Table 4.8 shows the selected recipes.

6. **Use the embedded web search capability to find recipes that are similar to the recipe you just chose (appetizer course and French cuisine). Choose one of the recipes returned from your web search. In addition to its name, please write down its source (for example, its URL or the name of the web site).**

   Task 6 asked the participants to use the embedded web search function to find recipes from the web that are similar to those chosen in Task 5. The participants could go through the result set of the query to find recipes they liked, annotate those web recipes against the facet space, and add them to the recipe collection. In experiment, all participants finished the task successfully. The recipes they chose are listed in Table 4.9, with a summary of sources listed in Table 4.10. Participants took an average of 124.65 seconds (standard deviation = 48.76).

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Shiitake mushroom appetizer | 1.0 | 6 | Vegetables | Quick | French, Appetizer |
| Beef roulades | 1.0 | 6 | Beef | Cook | French, Appetizer |
| Scallops with mushroom in white wine sauce | 1.0 | 3 | Vegetables | Bake | French, Appetizer |
| Chicken liver terrine | 1.0 | 2 | Chicken | Bake | French, Appetizer |
| Smoked salmon Rillettes | 1.0 | 1 | Fish | Cook | French, Appetizer |
| Foie gras toasts with greens and verjus port glaze | 1.0 | 1 | Vegetables | Bake | French, Appetizer, Winter |
| Artichokes in olive oil sauce | 1.0 | 1 | Vegetables | Cook | French, Appetizer |

Table 4.8: Recipes chosen in Task 5 with appetizer course from French cuisine.

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Mussels with Vegetables and Chardonnay Cream | 1.0 | 4 | Vegetables | Bake | French, Appetizer |
| Shrimp spread | 1.0 | 2 | Shrimp | Bake | French |
| Appetizer Pinata Meatballs | 1.0 | 1 | Beef | Cook | French |
| Bacon, cheese and olivermelt | 1.0 | 1 | Cheese | Bake | French |
| French Bread Appetizer | 1.0 | 1 | Cheese | Cook | French, Appetizer |
| Hors d'Asparagus | 1.0 | 1 | Cheese | Cook | French |
| Blue cheese bites | 1.0 | 1 | Cheese | Quick | French |
| Barras "Goey Choco" | 1.0 | 1 | Chocolate | Bake | French |
| Champagne chicken | 1.0 | 1 | Chicken | Bake | French |
| Overnight French Toast | 1.0 | 1 | Eggs | Cook | French |
| Jamican Grilled Fish | 1.0 | 1 | Fish | Grilled | French |
| Raisin-apricot glazed ham | 1.0 | 1 | Fruits | Cook | French |
| Appetizer kabobs | 1.0 | 1 | Potatoes | Grilled | French, Appetizer |
| Shrimp Cherchi | 1.0 | 1 | Shrimp | Quick | French |
| Broccoli/cauliflower casserole | 1.0 | 1 | Vegetables | Bake | French |
| French Zucchini Fritters | 1.0 | 1 | Vegetables | No Cook | French |

Table 4.9: Recipes chosen from the web with appetizer course and French cuisine as constraints in Task 6.

| Source | Frequency |
|---|---|
| frenchfood.about.com | 6 (2) |
| www.cooks.com | 6 (6) |
| www.recipezaar.com | 4 (4) |
| recipes.timerecordnews.com | 2 (2) |
| www.cookrecipes.com | 1 (1) |
| www.jamaicans.com | 1 (1) |

Table 4.10: Sources for recipes chosen in Task 6. The numbers of unique recipes from each site is indicated in parentheses.

7. **Imagine you want a recipe for the winter season. Choose at least 2 winter recipes that look interesting. Refine your query based on these selections and choose a recipe from the results.**

   Task 7 was intended to encourage the participants to explore the space of available recipes. Participants took an average of 102.8 seconds (standard deviation = 53.83). 18 of 20 participants completed the task successfully. However, 8 of 20 participants also asked for clarification about the question's request. The average score was 0.9 (standard deviation = 0.3). Table 4.11 shows the selected recipes.

### 4.2.3 Post-task

After finishing the tasks, participants completed an evaluation of the software. In general, the relationships between responses regarding the software usability, software helpfulness for exploring, software help for accessing alternatives, and web search option helpfulness are expected.

#### Likert-scale questions

Question 1 to Question 10 were Likert-scale questions. The participants gave subjective ratings of the system. The responses for the questions were summarized in Table 4.2.3. For each question, the frequencies for the different response categories were analyzed with the one-sample chi-square, or goodness-of-fit test.

| Recipe Name | Score | Times selected | Main ingredient | Method | Other facets |
|---|---|---|---|---|---|
| Chocolate-peppermint ice cream cake | 1.0 | 2 | Chocolate | Advance | American, Dessert, Winter |
| Chocolate hazelnut ginger biscotti | 1.0 | 1 | Chocolate | Bake | Italian, Dessert, Winter |
| Dry curried beans | 1.0 | 1 | Beans | Cook | Indian, Side Dish, Winter |
| Dulce de leche cheesecake squares | 1.0 | 1 | Cheese | Bake | Mexican, Dessert, Winter |
| Philippine-style chicken adobo | 1.0 | 1 | Chicken | Bake | Asia, Main course, Winter |
| Mango fool | 1.0 | 1 | Fruits | Cook | African, Dessert, Winter |
| Banana orange crepes | 1.0 | 1 | Fruits | Quick | French, Breakfast, Winter |
| Polvorones | 1.0 | 1 | Grains | Bake | French, Bread |
| Spice-crusted rack of lamb | 1.0 | 1 | Lamb | Roast | Indian, Main, Course, Winter |
| Jerk pork on red pepper mayo and black-eyed-pea cakes | 1.0 | 1 | Pork | Advance | American, Appetizer, Winter |
| Sesame wonton crisps | 1.0 | 1 | Pork | Advance | Asian, Main Winter |
| Hundred corner shrimp balls | 1.0 | 1 | Shrimp | Bake | Asian, Hors, Course, Winter |
| Spaghetti pie with broccoli rabe | 1.0 | 1 | Vegetables | Bake | Italian, Main Course, Winter |
| Braised pork with orange and fennel | 1.0 | 1 | Vegetables | Quick | American, Main Course, Winter |
| Warm chocolate tortes with raspberry sauce | 0.0 | 1 | Chocolate | Bake | Italian, Dessert |
| Chocolate almond torte | 0.0 | 1 | Chocolate | Bake | Italian, Dessert, Spring |
| Mango ice cream | 0.0 | 1 | Fruits | No cook | Asian, Dessert |
| Breakfast sausage casserole | 0.0 | 1 | Pork | Bake | American, Breakfast |
| Greek-style vegetable kebabs with orzo and feta | 0.0 | 1 | Vegetables | Grilled | Greek, Main Course, Summer |

Table 4.11: Recipes chosen in Task 7, using navigation functions of the system to find recipes for the winter season.

| | Task Question | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Participant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Avg. | SD |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 0.67 | 0 | 1 | 1 | 1 | 0.81 | 0.38 |
| 3 | 1 | 1 | 0.67 | 1 | 1 | 1 | 1 | 0.95 | 0.12 |
| 4 | 1 | 0.5 | 0.67 | 1 | 1 | 1 | 0 | 0.74 | 0.38 |
| 5 | 1 | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.93 | 0.19 |
| 6 | 1 | 0.5 | 0.67 | 1 | 1 | 1 | 1 | 0.88 | 0.21 |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 8 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 10 | 1 | 0.5 | 0.67 | 1 | 1 | 1 | 1 | 0.88 | 0.21 |
| 11 | 1 | 1 | 0.67 | 1 | 1 | 1 | 1 | 0.95 | 0.12 |
| 12 | 1 | 1 | 0.67 | 1 | 1 | 1 | 0 | 0.81 | 0.38 |
| 13 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 14 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 0.67 | 1 | 1 | 1 | 1 | 0.95 | 0.12 |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 17 | 1 | 0.5 | 0.67 | 1 | 1 | 1 | 1 | 0.88 | 0.21 |
| 18 | 1 | 1 | 0.67 | 1 | 1 | 1 | 1 | 0.95 | 0.12 |
| 19 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 20 | 1 | 0.5 | 1 | 1 | 1 | 1 | 1 | 0.92 | 0.19 |
| Avg. | 1 | 0.85 | 0.835 | 0.95 | 1 | 1 | 0.9 | X | X |
| SD | 0 | 0.24 | 0.17 | 0.22 | 0 | 0 | 0.3 | X | X |

Table 4.12: A summary of task scores, by task question and by participant.

The results indicated that the participants assigned positive ratings for the current software. The participants were satisfied with the software ($p < 0.01$). They indicated that the software functionality, including exploration and web search, was helpful to access the recipes. They also indicated that the usability of the software was good and that they would be very likely to use this software to find recipes in the future.

### Facet questions

From the perspective of the search interface designer, facet space is a very important concept both for annotating web data and searching for desired information within a collection. Participants were asked to rate the importance of each facet used in the study's facet space, namely `Cuisine`, `Course`, `Ingredient`, `Method`, `Season`, and `Special`. These ratings are shown in Table 4.15.

|  | Task Question |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Participant | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Avg. | SD |
| 1 | 90 | 100 | 62 | 23 | 36 | 79 | 74 | 58.12 | 34.76 |
| 2 | 85 | 86 | 58 | 59 | 58 | 124 | 58 | 66.25 | 34.76 |
| 3 | 65 | 42 | 69 | 61 | 118 | 90 | 84 | 66.5 | 34.19 |
| 4 | 46 | 54 | 54 | 31 | 43 | 245 | 92 | 71.13 | 74.44 |
| 5 | 19 | 18 | 85 | 29 | 31 | 96 | 52 | 41.88 | 33.01 |
| 6 | 45 | 121 | 32 | 36 | 39 | 216 | 112 | 75.88 | 69.37 |
| 7 | 14 | 27 | 21 | 14 | 15 | 133 | 48 | 34.88 | 41.57 |
| 8 | 80 | 49 | 108 | 40 | 46 | 114 | 150 | 74.38 | 47.04 |
| 9 | 21 | 44 | 23 | 15 | 24 | 121 | 53 | 38.75 | 36.30 |
| 10 | 74 | 35 | 72 | 37 | 53 | 95 | 235 | 76.38 | 69.43 |
| 11 | 13 | 54 | 30 | 55 | 62 | 169 | 42 | 54.50 | 50.05 |
| 12 | 21 | 58 | 51 | 29 | 25 | 150 | 92 | 54.75 | 46.29 |
| 13 | 50 | 42 | 33 | 26 | 29 | 92 | 109 | 49.25 | 33.76 |
| 14 | 36 | 38 | 16 | 48 | 21 | 111 | 92 | 47 | 35.96 |
| 15 | 19 | 38 | 45 | 29 | 31 | 97 | 184 | 57.25 | 57.20 |
| 16 | 67 | 43 | 68 | 42 | 57 | 185 | 168 | 80.75 | 61.55 |
| 17 | 68 | 48 | 60 | 49 | 43 | 140 | 163 | 73.50 | 50.73 |
| 18 | 24 | 57 | 67 | 25 | 41 | 59 | 84 | 46.88 | 23.61 |
| 19 | 24 | 25 | 26 | 48 | 20 | 118 | 36 | 39.5 | 33.12 |
| 20 | 19 | 50 | 53 | 38 | 36 | 59 | 128 | 50.38 | 34.59 |
| Avg. | 44 | 51.45 | 51.65 | 36.7 | 41.4 | 124.65 | 102.8 | X | X |
| SD | 26.40 | 24.97 | 23.69 | 13.58 | 22.42 | 48.76 | 53.83 | X | X |

Table 4.13: A summary of task times, by task question and by participant.

The results indicate that ingredient was a very important facet for recipe search ($p < 0.01$). The method facet was ranked in second place, but only slightly better rated than cuisine and course. The facets ingredient and method were thereby chosen to organize the distribution of recipes within the database ( see Table 4.2). Participants indicated that the facet special was unimportant ($p < 0.01$).

Question 12 asked participants if there are any other attributes for recipes. 8 participants thought the facets provided by the interface were enough to describe a recipe. Although the interface displayed the detailed recipe (including serving size, cooking time, amount of ingredient as well as nutrition information), some individuals felt that those attributes were important when exploring and evaluating cooking recipes. The attributes “Other ingredient” and “Cooking skills” were also identified as potential facets.

| Topic | Response Category | Results |
|---|---|---|
| Q1. Feeling about software | Very unsatisfied | 0 |
| | Unsatisfied | 0 |
| | Satisfied | 13 |
| | Very satisfied | 7 |
| Q2. Number of alternative recipes | Very small | 0 |
| | Small | 1 |
| | Large | 13 |
| | Very large | 6 |
| Q3. Accessibility of alternative recipes | Very poor | 0 |
| | Poor | 1 |
| | Good | 12 |
| | Very Good | 6 |
| Q4. Helpfulness of software for accessing alternative recipes | Very unhelpful | 0 |
| | Unhelpful | 1 |
| | Helpful | 11 |
| | Very helpful | 8 |
| Q5. Helpfulness of exploration | Very unhelpful | 0 |
| | unhelpful | 2 |
| | Helpful | 13 |
| | Very helpful | 5 |
| Q6. Helpfulness of the web search option | Very unhelpful | 0 |
| | Unhelpful | 6 |
| | Helpful | 13 |
| | Very helpful | 1 |
| Q7. Helpfulness of software for viewing chosen recipes | Very unhelpful | 0 |
| | Unhelpful | 1 |
| | Helpful | 8 |
| | Very helpful | 11 |
| Q8. Usability of software | Very poor | 0 |
| | Poor | 1 |
| | Good | 10 |
| | Very Good | 9 |
| Q9. Likelihood of using software to find recipes | Very unlikely | 0 |
| | Unlikely | 1 |
| | Likely | 7 |
| | Very likely | 12 |
| Q10. Software compared to other means | Much worse | 0 |
| | Worse | 1 |
| | Better | 8 |
| | Much better | 11 |

Table 4.14: The Response of the post-task questionnaire for questions 1 through 10, N = 20. 1 means "Strongly Disagree," 4 means "Strongly Agree." Missing responses were coded with 0.

| Facet | Very<br>Unimportant | Unimportant | Important | Very<br>Important | Score | Rank |
|---|---|---|---|---|---|---|
| Cuisine | 1 | 4 | 9 | 6 | 60 | 3 |
| Course | 1 | 4 | 9 | 6 | 60 | 3 |
| Ingredient | 0 | 1 | 7 | 12 | 71 | 1 |
| Method | 0 | 5 | 9 | 6 | 61 | 2 |
| Special | 0 | 11 | 8 | 1 | 50 | 5 |

Table 4.15: The response and analysis of facets in the post-task questionnaire for questions 11. N = 20

### Open-ended questions

The detailed answers for the open-ended Questions 13 to 18 are posted in Appendix D.

All participants were satisfied to explore and evaluate cooking recipes with the interface. 19 of 20 participants said the interface was helpful to choose the recipes they want. They prefer the recipe search accompanied with obvious semantics and images, which helps them decide where to look. This was echoed in the following comments: "The pictures of the recipe, ability to see the recipe without losing the search" and "The pictures really helped with choosing recipes because I did not know what many of the recipes were just by name." Participants exhibited strong positive feelings toward the system, with one participant saying "the picture showing the dish is good, the search tool (create new space) seems to be fast and effective."

Participants found it easier to refine and expand their searches using the "New" space button in the interface, and arrange result sets in multiple ways using "view key," which "allows you to select what preference you are looking for." Some participants' comments about the interface support this conclusion:

> The visual searching of recipes is handy compared to a text search
>
> I like the idea of this software a lot. It would help me getting rid of ingredients. I think the layout is good, and once learned, very fast, powerful and easy to use.

> To see the recipe visually with a title, the ability to restrict into broad categories
>
> I enjoyed the visual representation of the food and the searching with selected options ability
>
> The picture showing the dish is good, the search tool (create new space) seems to be fast and effective
>
> Being able to go back on query, the way you can decide what cuisine/dish, etc. you want
>
> Search function, it was easy to use and quick

However, some shortcomings of the interface became apparent in the experiment, mainly related to "web search" and the meanings of some labels. Participants commented:

> The web search had trouble with the window size (or maybe source website did), web search was difficult to understand and use.
>
> Web search, the html didn't render properly and I like to see the list of results first, then choose the recipe, rather than just go to the first page
>
> Web search was not very useful as I would just resort to using my own web browser and search (Google).

The interface does have a drawback that is not usually encountered in web search. In Java technology, there is some limitation for the JTextPane panel to display the HTML source. Thus, certain HTML source code crashes this panel and can not be rendered properly, which results in garbage words being displayed on the screen.

Some participants gave suggestions for the interface, and these open issues will be considered for future work:

- The new space dialogue could be better if it adopted pull down menus.

- When flipping pages, the space visually keeps selection.
- An option to store some of your favorite recipes so that you can find them faster next time.
- Language support.
- Print out recipe in a compact form
- There should be the option for adding data to collection in the web search panel, such as a link that says: "would you like to store this search information?" When the user clicks on this link a window will pop up with fields to fill in the information to store.

Reflecting on the results of the experiment, participants were able to successfully complete the tasks in a finite time, and they reached an appropriate set of recipes in the collection.

# Chapter 5

# Conclusions

The design and implementation of a metadata-based prototype system for the annotation and retrieval of web-based information is the main contribution of this thesis. The implemented system is interactive and extensible, and it can be applied to any web-based information resource.

## 5.1 Summary of Contributions

This work discussed the main aspects of the technical realization for the metadata-based prototype system. The system focuses on two main aspects:

**Annotating web-based recipes:** A facet space, derived from faceted metadata, is built to describe the recipe domain from different aspects. It helps the user understand the content and structure of the recipes and provides a standard metadata to annotate web-based recipes. The facets initially identified are "Cuisine, Course, Ingredient, Method, Season, and Special." More details including the possible values for each facet are given in Table 4.1. From the usability study, more potential attributes were identified, including: serving size, cooking time, amount of ingredient, nutrition information, as well as relative difficulty of the recipe. The trained user can extend the space by adding, editing, and reusing present facets without disturbing the whole structure.

XFML format was chosen as a language for encoding the facet space in our application because it is intended for the description of faceted metadata. It explicitly expresses the meaning of the web resources and enables machines to execute tasks like annotating web-based recipes, searching for recipes via query specification, and flexible matching at the semantic level.

**Retrieving recipes based on content:** The system provides a retrieval environment for the user to find recipes of interest. Based on the integrated facet space, the user can browse the semantic data together with the original web pages and see the context of the data, make judgements about them based on his or her own criteria, and iteratively query the data until she finds recipes of interest. To avoid dead ends or empty result sets, a similarity analysis is introduced, which benefits the user by allowing sets of recipes to be arranged according to their semantic distances. The system also integrates search functions of web search engines (such as Google) and returns a set of similar web-based recipes with URL addresses.

After several rounds of interface redesign, a usability study was conducted to evaluate the current prototype system. Twenty participants, recruited from the Department of Computer Science participant pool, were asked to complete a series of tasks to evaluate access to a recipe collection. These participants found that the system was helpful for browsing tasks, and they exhibited strong positive feelings about the interface. Success with design goals was echoed by comments from participants such as "the software was helpful in helping me choose the recipes I want" and "It did a reasonably good job of guiding me through the selection process."

## 5.2 Future Work

Guidelines for system design modification are derived from the user feedback. After completing this project, several areas have been identified to increase the capabilities of the system:

- Use data mining technologies [23, 40] to infer and generate semantic metadata for an application domain.

- Test the system with other sample applications, such as facial photo manipulation [33].

- Put the system on the web, with technologies such as PHP [48], so that it can be accessed by more diverse people, leading to wider use and better feedback.

- Include support for more complex web content descriptions and a richer vocabulary for the user to describe his or her conceptual space.

- Design and implement a recommendation system that employs past user experiences to aid new user navigation and search. Data mining technologies could be used to realize it.

- Infer the semantic attributes of a conceptual space with the information given by the assessment of each user at the beginning of each session.

- Guide users to under-explored parts of the space. For example, in the user study, no one selected recipes with ingredients of rice or noodles, or cooking methods of broil or steam.

# Bibliography

[1] H. Abrams, K. Watsen, and M. Zyda. Three-tiered interest management for large-scale virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 125–129, New York, NY, USA, 1998. ACM Press.

[2] M. Baca and A. Gilliland-Swetland. *Introduction to Metadata: Pathways to Digital Information.* Getty Publications, 2000.

[3] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language Reference W3C recommendation. Online: [http://www.w3.org/TR/2004/REC-owl-ref-20040210/], Retrieved 2004.

[4] D. Beckett. RDF/XML syntax specification (revised) W3C recommendation. Online: [http://www.w3.org/TR/rdf-syntax-grammar/], Retrieved 2004.

[5] J. Berlin and A. Motro. Database schema matching using machine learning with feature selection. In *Proceedings of the 14th International Conference on Advanced Information Systems Engineering*, pages 452–466, London, UK, 2002. Springer-Verlag.

[6] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 184(5):34–43, 2001.

[7] J. S. Bowman, S. L. Emerson, and M. Darnovsky. *The Practical SQL Handbook.* Addison Wesley Professional, 2001.

[8] T. Bray, D. Hollander, A. Layman, and R. Tobin. Namespaces in XML 1.1 W3C Recommendation. Online: [http://www.w3.org/TR/2004/REC-xml-names11-20040204/], Retrieved 2004.

[9] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible Markup Language (XML) 1.0 (third edition) W3C. Online: [http://www.w3.org/TR/2004/REC-xml-20040204/], Retrieved 2004.

[10] D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF Schema W3C Recommendation. Online: [http://www.w3.org/TR/2004/REC-rdf-schema-20040210/], Retrieved 2004.

[11] B. Chang, J. Kesselman, and R. Rahman. Document object model (DOM) level 3 validation specification version 1.0 W3C recommendation. Online: [http://www.w3.org/TR/2004/REC-DOM-Level-3-Val-20040127/], Retrieved 2004.

[12] H. Chen, T. Finin, A. Joshi, F. Perich, D. Chakraborty, and L. Kagal. Intelligent Agents Meet the Semantic Web in Smart Spaces. *IEEE Internet Computing*, 8(6), November 2004.

[13] B. Cole, R. Eckstein, J. Elliott, M. Loy, and D. Wood. *Java Swing*. O'Reilly & Associates, Inc., 2002.

[14] D. Connolly, F. van Harmelen, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. A. Stein. Annotated DAML+OIL Ontology Markup W3C. Online: [http://www.w3.org/TR/daml+oil-walkthru/], Retrieved 2001.

[15] DCMI. Dublin Core Metadata Initiative. Online: [http://dublincore.org/], Retrieved 2004.

[16] W. Denton. How to make a faceted classification and put it on the web. Online: [http://www.miskatonic.org/library/facet-web-howto.html], Retrieved 2003.

[17] P. Van Dijck. XFML core - eXchangeable Faceted Metadata Language. Online: [http://www.xfml.org/spec/1.0.html], Retrieved 2003.

[18] P. Dubois. *MySQL Cookbook.* O'Reilly & Associates, Inc., 2001.

[19] A. Elliott. Flamenco image browser: using metadata to improve image search during architectural design. In *CHI '01 extended abstracts on Human factors in computing systems*, pages 69–70, New York, NY, USA, 2001. ACM Press.

[20] D.C. Fallside. XML Schema Part 0: Primer, W3C Recommendation W3C. Online: [http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/], Retrieved 2001.

[21] Google. Google Web APIs. Online: [http://www.google.com/apis/], Retrieved 2004.

[22] B. V. Haecke. *JDBC: Java Database Connectivity.* Wiley Publishing, 1997.

[23] J. Han and M. Kamber. *Data Mining : Concepts and Techniques.* Morgan Kaufmann, 2000.

[24] D. H. Hepting. Towards a visual interface for information visualization. In E. Banissi, editor, *Proceedings of the Sixth International Conference on Information Visualization*, pages 295–302. IEEE Computer Society, 2002.

[25] E. Hyvönen, S. Saarela, and K. Viljanen. Application of ontology techniques to view-based semantic search and browsing. In *ESWS*, pages 92–106, 2004.

[26] J. R. Jackson and A. L. McClellan. *JAVA by Example.* Sun Microsystems, Inc., 1999.

[27] K. Jarvelin, J. Kekalainen, and T. Niemi. Expansiontool: Concept-based query expansion and construction. *Information Retrieval,*, 4(3/4):231–255, 2001.

[28] D. R. Karger and D. Quan. Haystack: a user interface for creating, browsing, and organizing arbitrary semistructured information. In *CHI '04 extended abstracts on Human factors in computing systems*, pages 777–778, New York, NY, USA, 2004. ACM Press.

[29] M. S. Lacher and G. Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 305–309. AAAI Press, 2001.

[30] A. Laender, B. Ribeiro-Neto, A. Silva, and J. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31:84–93, June 2002.

[31] W. Li and C. Clifton. Semint: a tool for identifying attribute correspondences in heterogeneous databases using neural networks. *Data Knowl. Eng.*, 33(1):49–84, 2000.

[32] M. Markkula and E. Sormunen. End-user searching challenges indexing practices inthe digital newspaper photo archive. *Inf. Retr.*, 1(4):259–285, 2000.

[33] Sun Microsystems. Java web start technology. Online: [http://java.sun.com/products/javawebstart/], Retrieved 2005.

[34] J. Pan and I. Horrocks. Metamodeling architecture of web ontology languages. In *Proceedings of the Semantic Web Working Symposium*, pages 131–149, July 2001.

[35] W. Pratt, M. A. Hearst, and L. M. Fagan. A knowledge-based approach to organizing retrieved documents. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 80–85, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[36] U. Priss. Faceted information representation. In *Proceedings of the 8th International Conference on Conceptual Structures*, pages 84–94, citeseer.nj.nec.com/priss00faceted.html, 2000. Shaker Verlag, Aachen.

[37] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[38] Vividence Research. Tangled web. Online: [http://www.vividence.com], Retrieved 2001.

[39] G. Salton and M.J.McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, New York, 1983.

[40] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.

[41] B. Shneiderman, D. Feldman, A. Rose, and X. Grau. Visualizing digital library search results with categorical and hierarchical axes. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 57–66, San Antonio TX, 2000.

[42] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts.* McGraw-Hill Book Company, 3rd edition, 1997.

[43] The Internet Society. Uniform Resource Identifier (URI): Generic Syntax. Online: [http://www.w3.org/Addressing/], Retrieved 2005.

[44] Siderean Software. From site search to the semantic web:a white paper, Retrieved 2003.

[45] D. Sullivan. Search engine sizes, Retrieved 2005.

[46] Y. Tzitzikas, N. Spyratos, P. Constantopoulos, and A. Analyti. Extended faceted taxonomies for web catalogs. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pages 192–204, Washington, DC, USA, 2002. IEEE Computer Society.

[47] D. Weiss. Introduction to search results clustering. In *Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference*, pages 209–218, Rzeszów, Poland, 2002.

[48] H. E. Williams and D. Lane. *Web Database Applications with PHP & MySQL.* O'Reilly & Associates, Inc., 2002.

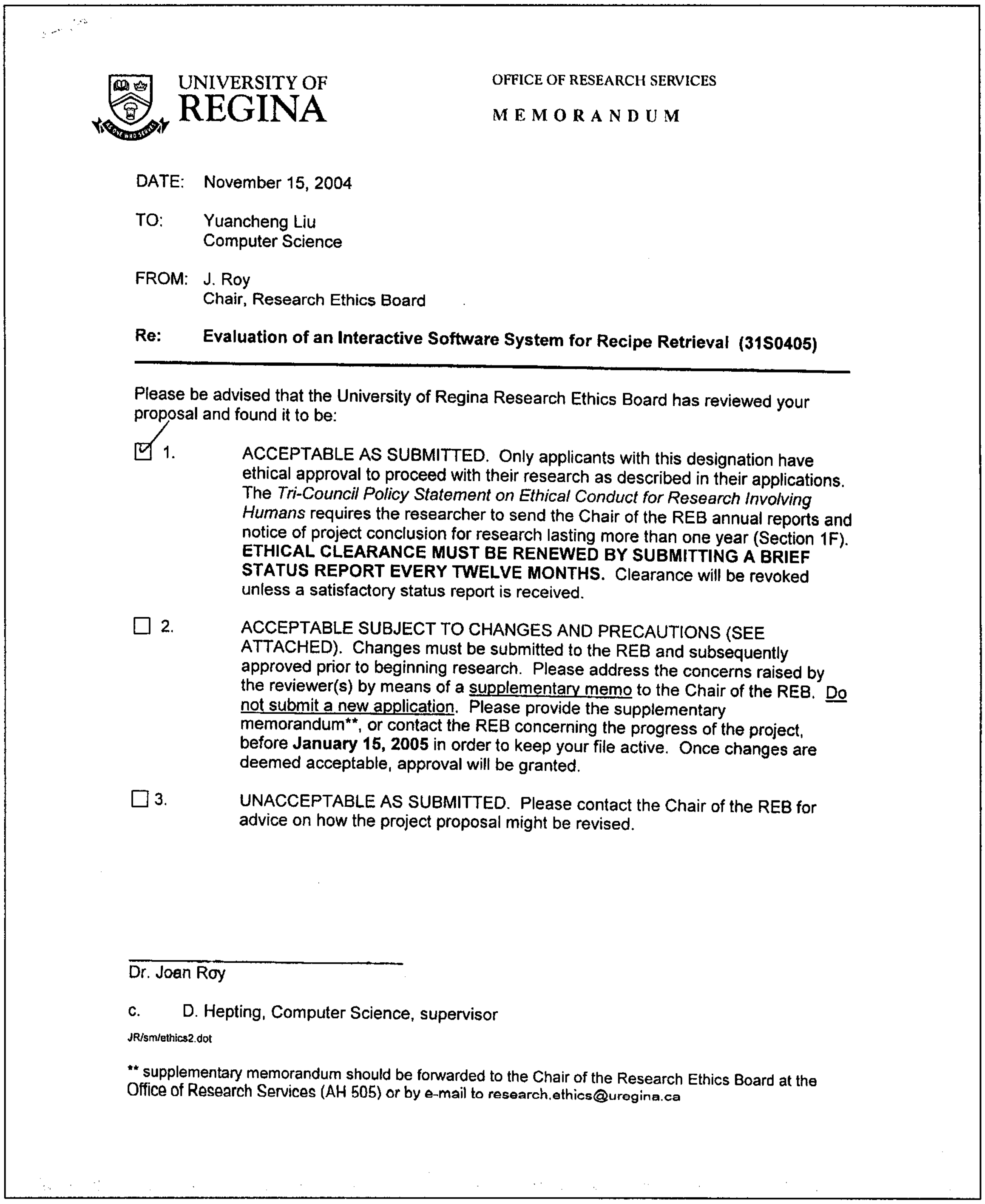
[49] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003. ACM Press.

[50] B. Yuwono and D. L. Lee. Search and ranking algorithms for locating resources on the world wide web. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 164–171, Washington, DC, USA, 1996. IEEE Computer Society.

# Appendix A

# Ethics Approval

This appendix contains a copy of the ethics approval memo from the University of Regina Research Ethics Board (REB).



UNIVERSITY OF
REGINA

OFFICE OF RESEARCH SERVICES

MEMORANDUM

DATE: November 15, 2004

TO: Yuancheng Liu
Computer Science

FROM: J. Roy
Chair, Research Ethics Board

**Re: Evaluation of an Interactive Software System for Recipe Retrieval (31S0405)**

Please be advised that the University of Regina Research Ethics Board has reviewed your proposal and found it to be:

☑ 1. ACCEPTABLE AS SUBMITTED. Only applicants with this designation have ethical approval to proceed with their research as described in their applications. The *Tri-Council Policy Statement on Ethical Conduct for Research Involving Humans* requires the researcher to send the Chair of the REB annual reports and notice of project conclusion for research lasting more than one year (Section 1F). **ETHICAL CLEARANCE MUST BE RENEWED BY SUBMITTING A BRIEF STATUS REPORT EVERY TWELVE MONTHS.** Clearance will be revoked unless a satisfactory status report is received.

□ 2. ACCEPTABLE SUBJECT TO CHANGES AND PRECAUTIONS (SEE ATTACHED). Changes must be submitted to the REB and subsequently approved prior to beginning research. Please address the concerns raised by the reviewer(s) by means of a supplementary memo to the Chair of the REB. Do not submit a new application. Please provide the supplementary memorandum**, or contact the REB concerning the progress of the project, before **January 15, 2005** in order to keep your file active. Once changes are deemed acceptable, approval will be granted.

□ 3. UNACCEPTABLE AS SUBMITTED. Please contact the Chair of the REB for advice on how the project proposal might be revised.

Dr. Joan Roy

c. D. Hepting, Computer Science, supervisor

JR/sm/ethics2.dot

** supplementary memorandum should be forwarded to the Chair of the Research Ethics Board at the Office of Research Services (AH 505) or by e-mail to research.ethics@uregina.ca

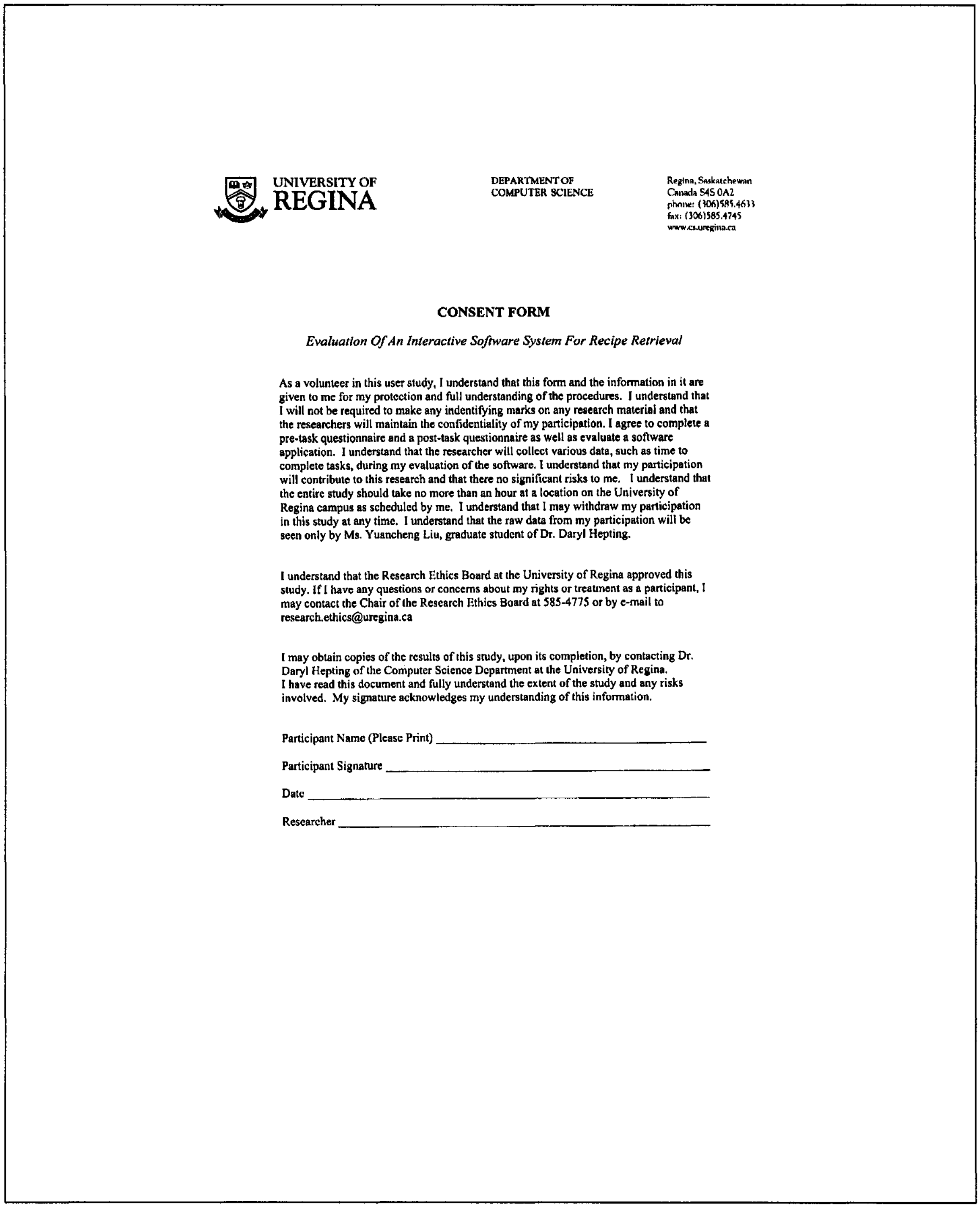
Figure A.1: Ethics approval

# Appendix B

# User Study Materials

This appendix contains materials involved in the test and experiment entitled "Evaluation of an Interactive Software System for Recipe Retrieval." Materials are presented in the following order:

- Consent form
- Pre-task questionnaire
- Tutorial
- Task questions
- Post-task questionnaire



UNIVERSITY OF
REGINA

DEPARTMENT OF
COMPUTER SCIENCE

Regina, Saskatchewan
Canada S4S 0A2
phone: (306)585.4633
fax: (306)585.4745
www.cs.uregina.ca

**CONSENT FORM**

*Evaluation Of An Interactive Software System For Recipe Retrieval*

As a volunteer in this user study, I understand that this form and the information in it are given to me for my protection and full understanding of the procedures. I understand that I will not be required to make any indentifying marks on any research material and that the researchers will maintain the confidentiality of my participation. I agree to complete a pre-task questionnaire and a post-task questionnaire as well as evaluate a software application. I understand that the researcher will collect various data, such as time to complete tasks, during my evaluation of the software. I understand that my participation will contribute to this research and that there no significant risks to me. I understand that the entire study should take no more than an hour at a location on the University of Regina campus as scheduled by me. I understand that I may withdraw my participation in this study at any time. I understand that the raw data from my participation will be seen only by Ms. Yuancheng Liu, graduate student of Dr. Daryl Hepting.

I understand that the Research Ethics Board at the University of Regina approved this study. If I have any questions or concerns about my rights or treatment as a participant, I may contact the Chair of the Research Ethics Board at 585-4775 or by e-mail to research.ethics@uregina.ca

I may obtain copies of the results of this study, upon its completion, by contacting Dr. Daryl Hepting of the Computer Science Department at the University of Regina.
I have read this document and fully understand the extent of the study and any risks involved. My signature acknowledges my understanding of this information.

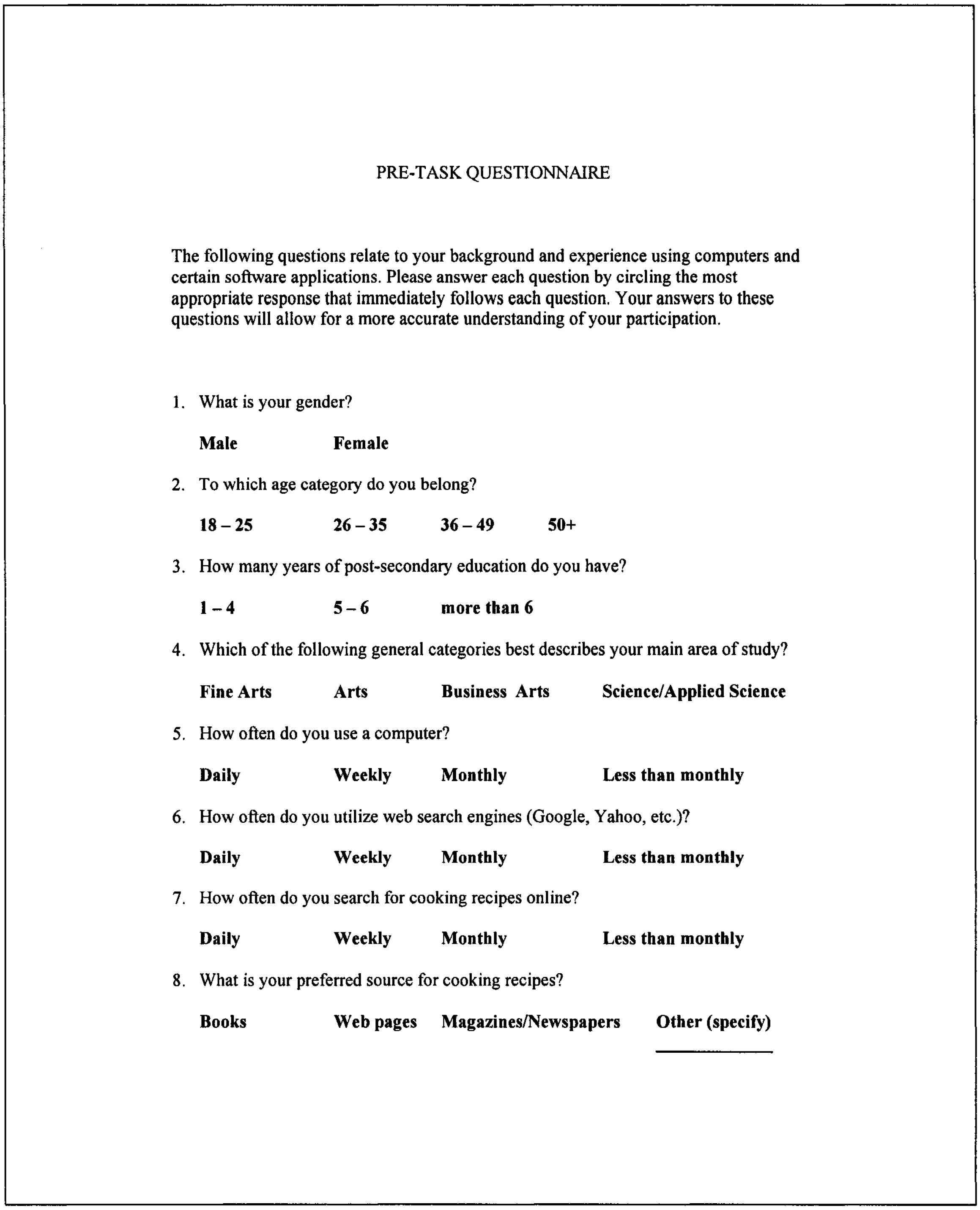
Participant Name (Please Print) ______________________________

Participant Signature ______________________________

Date ______________________________

Researcher ______________________________

Figure B.1: Consent Form



PRE-TASK QUESTIONNAIRE

The following questions relate to your background and experience using computers and certain software applications. Please answer each question by circling the most appropriate response that immediately follows each question. Your answers to these questions will allow for a more accurate understanding of your participation.

1. What is your gender?

   **Male** **Female**

2. To which age category do you belong?

   **18 – 25** **26 – 35** **36 – 49** **50+**

3. How many years of post-secondary education do you have?

   **1 – 4** **5 – 6** **more than 6**

4. Which of the following general categories best describes your main area of study?

   **Fine Arts** **Arts** **Business Arts** **Science/Applied Science**

5. How often do you use a computer?

   **Daily** **Weekly** **Monthly** **Less than monthly**

6. How often do you utilize web search engines (Google, Yahoo, etc.)?

   **Daily** **Weekly** **Monthly** **Less than monthly**

7. How often do you search for cooking recipes online?

   **Daily** **Weekly** **Monthly** **Less than monthly**

8. What is your preferred source for cooking recipes?

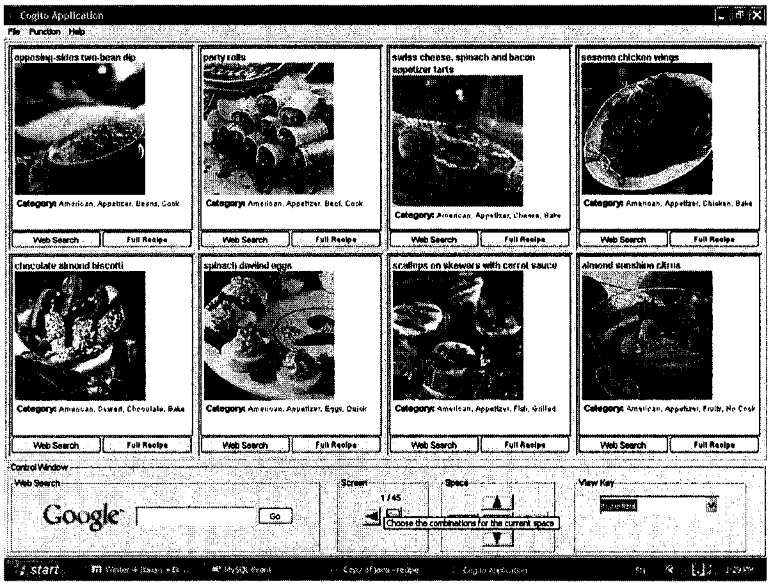
   **Books** **Web pages** **Magazines/Newspapers** **Other (specify)** ____________

Figure B.2: Pre-task Questionnaire

**Using the software:**

***Please try the software as you read this introduction – and ask any questions you may have.***

At any time you will see at most 8 recipes on your display. For each recipe, you see its name, possibly a photograph of the finished dish, and the categories to which this recipe belongs. Below each recipe, you see 2 buttons. The "Web Search" button allows you to initiate a web search for recipes that might be categorized in the same way as the one displayed. The "Full Recipe" button shows the full text of the recipe if it is available, otherwise it gives a link to the recipe's website. You may select a recipe (for further consideration by clicking on it.



opposing-sides two-bean dip
**Category:** American, Appetizer, Beans, Cook

party rolls
**Category:** American, Appetizer, Beef, Cook

swiss cheese, spinach and bacon appetizer tarts
**Category:** American, Appetizer, Cheese, Bake

sesame chicken wings
**Category:** American, Appetizer, Chicken, Bake

chocolate almond biscotti
**Category:** American, Dessert, Chocolate, Bake

spinach deviled eggs
**Category:** American, Appetizer, Eggs, Quick

scallops on skewers with carrot sauce
**Category:** American, Appetizer, Fish, Grilled

almond sunshine citrus
**Category:** American, Appetizer, Fruits, No Cook

Screen: 1 / 45

Choose the combinations for the current space

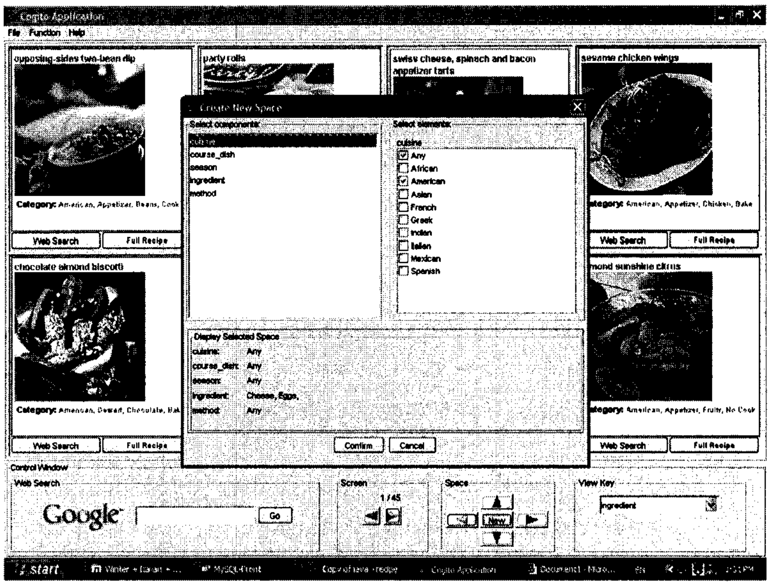
At the bottom of the display (shown above), the "Screen" control allows you to view more screens of recipes. In this example, you are seeing Screen 1 of 45 ( which contains 8 of approximately 260 recipes).

To the right of the "Screen" control is the "Space" control. At the centre of this control is the "New" button, which allows you to determine which recipes you will see next. The dialogue box which appears when you click this button will show you which elements (from each component) appear in your selected recipes. The details of this dialogue box are described below.

Figure B.3: Tutorial for the Study, page 1

To the right of the "Space" Control is the "View Key" control. This control allows you to select how the recipes are arranged. In the example above, the View Key is set to "ingredient" and each recipe in the display uses a different ingredient. In the example, the 45 screens of recipes show a sample recipe using every included ingredient, then another set of samples and another set of samples until all recipes have been shown. This gives you a sense of what is available within the current recipes without having to see all of them.

The screenshot below illustrates what happens when the "New" button is pressed. Since "ingredient" is the View Key, the default action (if you would click "Confirm" on the dialogue box without doing anything else) would be to next show you all recipes with ingredients of "cheese" or "eggs." Notice the 3 areas of this dialogue box: "Select components" shows you the different ways in which the recipes are categorized. In this example, you may explore recipes based on their "cuisine", "course/dish", "season", "ingredient", or "method" (of cooking). Clicking on any of the component names will show you which elements or values have been selected for that component (under the "Select elements" heading, to the right of "Select components"). Below these two areas is "Display Selected Space" that lists the values for every component.



Create New Space

Select components: cuisine, course_dish, season, ingredient, method

Select elements:

cuisine
☑ Any
□ African
☑ American
□ Asian
□ French
□ Greek
□ Indian
□ Italian
□ Mexican
□ Spanish

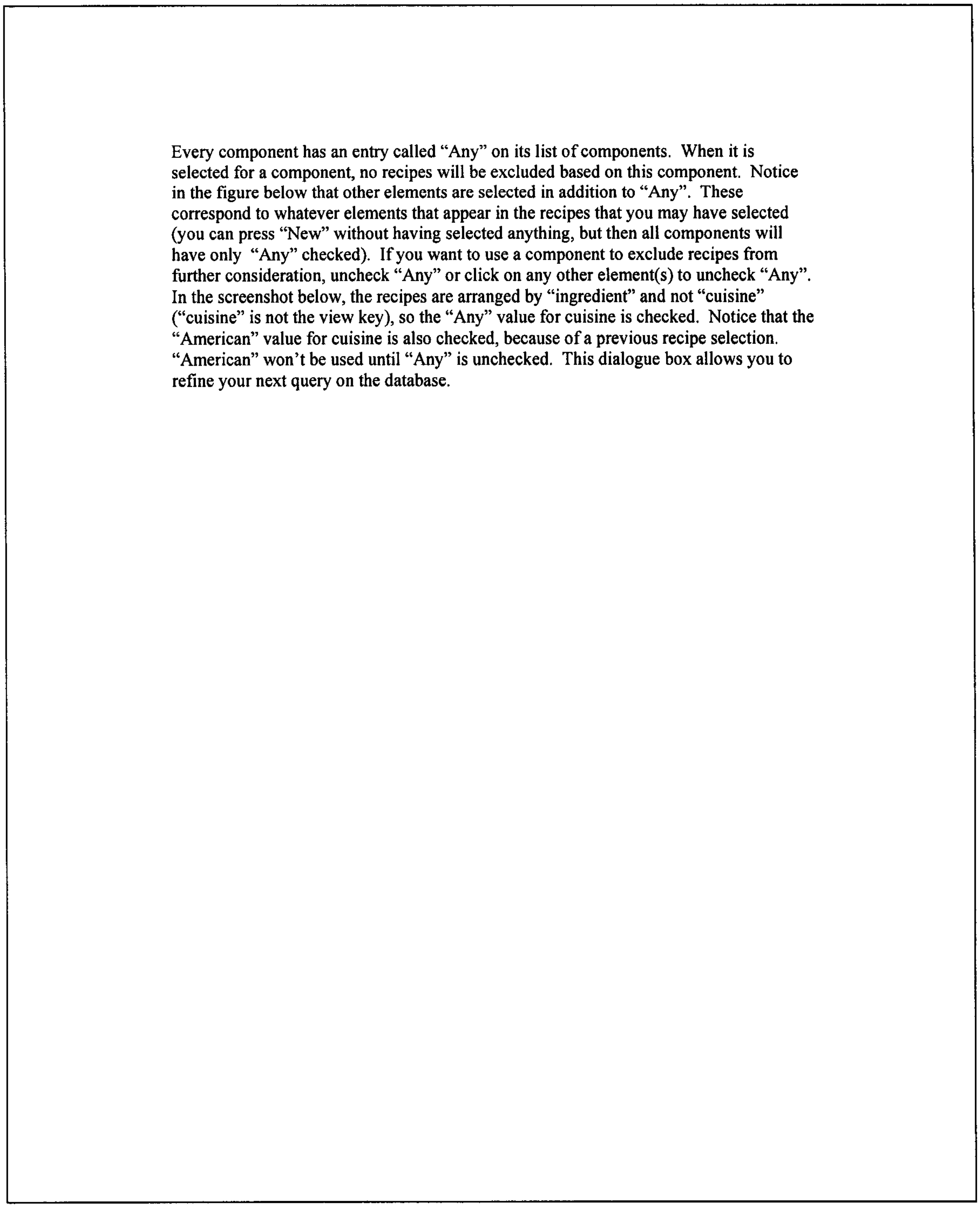
Display Selected Space
cuisine: Any
course_dish: Any
season: Any
ingredient: Cheese, Eggs,
method: Any

Screen: 1/45

View Key: ingredient

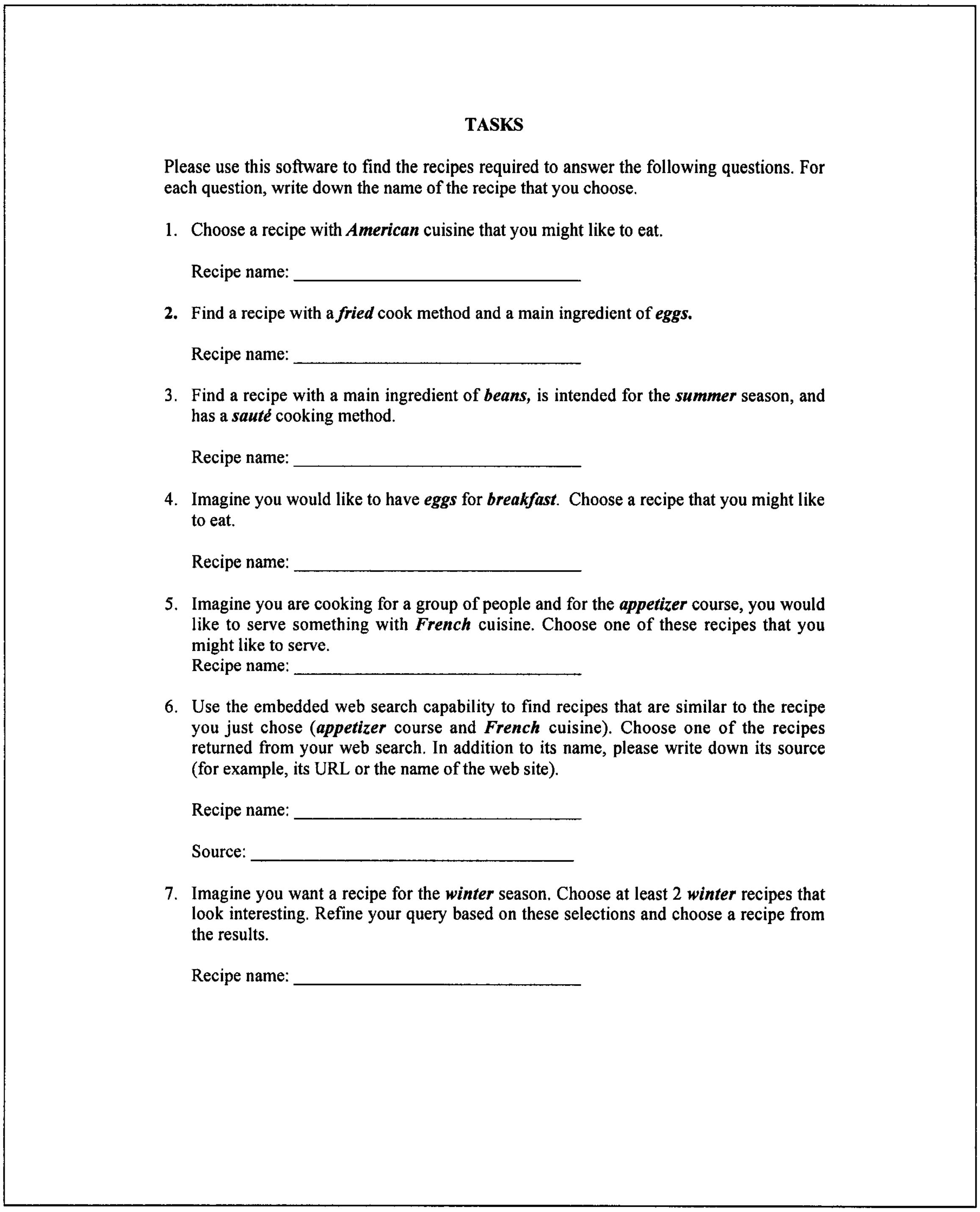
Cogito Application
File Function Help
opposing-sides two-bean dip
party rolls
swiss cheese, spinach and bacon appetizer tarts
sesame chicken wings
Category: American, Appetizer, Beans, Cook
Category: American, Appetizer, Chicken, Bake
Web Search
Full Recipe
chocolate almond biscotti
mond sunshine citrus
Category: American, Dessert, Chocolate, Bak
Category: American, Appetizer, Fruit, No Cook
Confirm
Cancel
Control Window
Web Search
Google
Go
Space
New
start

Figure B.4: Tutorial for the Study, page 2



Every component has an entry called "Any" on its list of components. When it is selected for a component, no recipes will be excluded based on this component. Notice in the figure below that other elements are selected in addition to "Any". These correspond to whatever elements that appear in the recipes that you may have selected (you can press "New" without having selected anything, but then all components will have only "Any" checked). If you want to use a component to exclude recipes from further consideration, uncheck "Any" or click on any other element(s) to uncheck "Any". In the screenshot below, the recipes are arranged by "ingredient" and not "cuisine" ("cuisine" is not the view key), so the "Any" value for cuisine is checked. Notice that the "American" value for cuisine is also checked, because of a previous recipe selection. "American" won't be used until "Any" is unchecked. This dialogue box allows you to refine your next query on the database.

Figure B.5: Tutorial for the Study, page 3



**TASKS**

Please use this software to find the recipes required to answer the following questions. For each question, write down the name of the recipe that you choose.

1. Choose a recipe with ***American*** cuisine that you might like to eat.

   Recipe name: ______________________________

2. Find a recipe with a ***fried*** cook method and a main ingredient of ***eggs.***

   Recipe name: ______________________________

3. Find a recipe with a main ingredient of ***beans,*** is intended for the ***summer*** season, and has a ***sauté*** cooking method.

   Recipe name: ______________________________

4. Imagine you would like to have ***eggs*** for ***breakfast***. Choose a recipe that you might like to eat.

   Recipe name: ______________________________

5. Imagine you are cooking for a group of people and for the ***appetizer*** course, you would like to serve something with ***French*** cuisine. Choose one of these recipes that you might like to serve.
   Recipe name: ______________________________

6. Use the embedded web search capability to find recipes that are similar to the recipe you just chose (***appetizer*** course and ***French*** cuisine). Choose one of the recipes returned from your web search. In addition to its name, please write down its source (for example, its URL or the name of the web site).

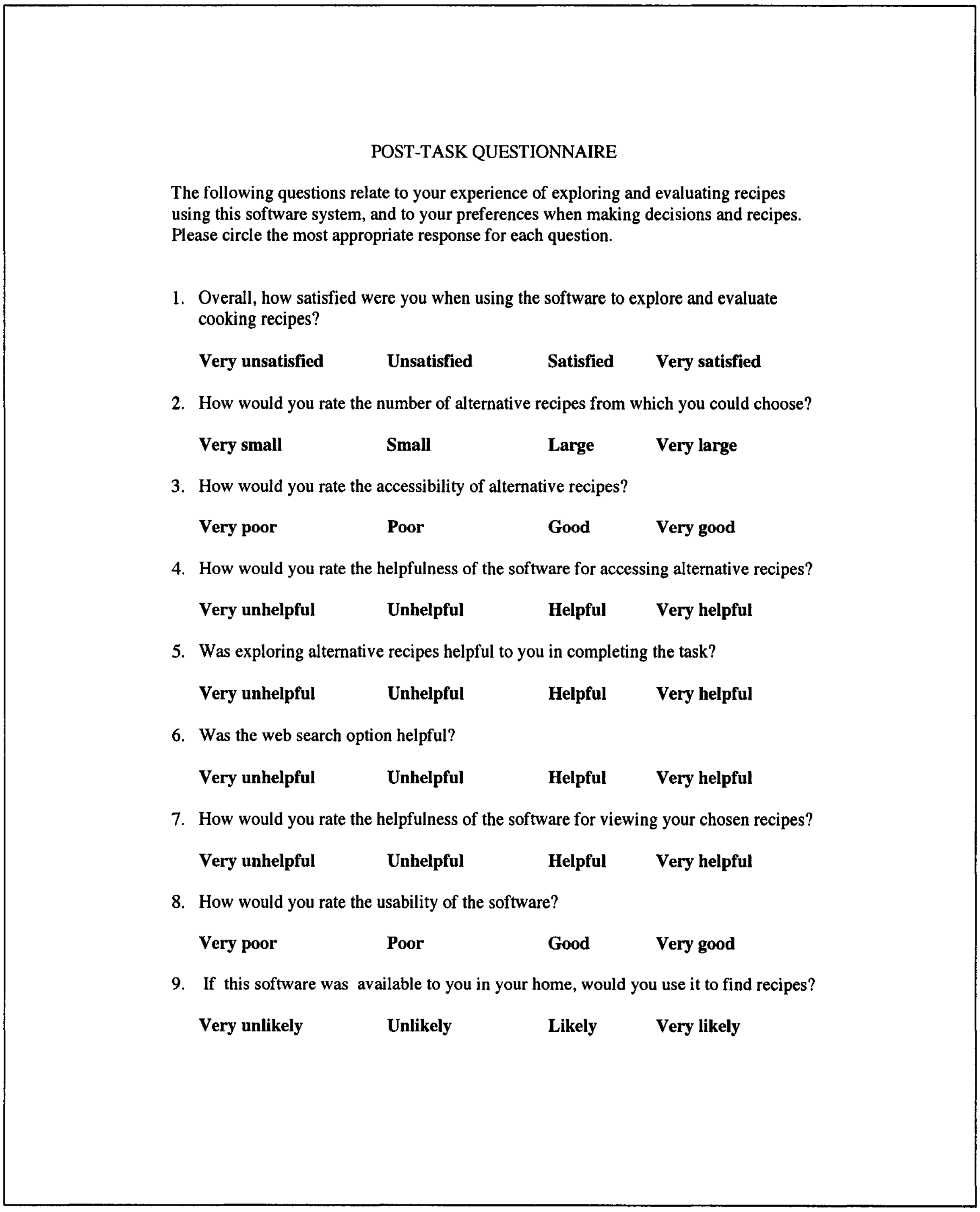
   Recipe name: ______________________________

   Source: __________________________________

7. Imagine you want a recipe for the ***winter*** season. Choose at least 2 ***winter*** recipes that look interesting. Refine your query based on these selections and choose a recipe from the results.

   Recipe name: ______________________________

Figure B.6: Tasks



POST-TASK QUESTIONNAIRE

The following questions relate to your experience of exploring and evaluating recipes using this software system, and to your preferences when making decisions and recipes. Please circle the most appropriate response for each question.

1. Overall, how satisfied were you when using the software to explore and evaluate cooking recipes?

   **Very unsatisfied** **Unsatisfied** **Satisfied** **Very satisfied**

2. How would you rate the number of alternative recipes from which you could choose?

   **Very small** **Small** **Large** **Very large**

3. How would you rate the accessibility of alternative recipes?

   **Very poor** **Poor** **Good** **Very good**

4. How would you rate the helpfulness of the software for accessing alternative recipes?

   **Very unhelpful** **Unhelpful** **Helpful** **Very helpful**

5. Was exploring alternative recipes helpful to you in completing the task?

   **Very unhelpful** **Unhelpful** **Helpful** **Very helpful**

6. Was the web search option helpful?

   **Very unhelpful** **Unhelpful** **Helpful** **Very helpful**

7. How would you rate the helpfulness of the software for viewing your chosen recipes?

   **Very unhelpful** **Unhelpful** **Helpful** **Very helpful**

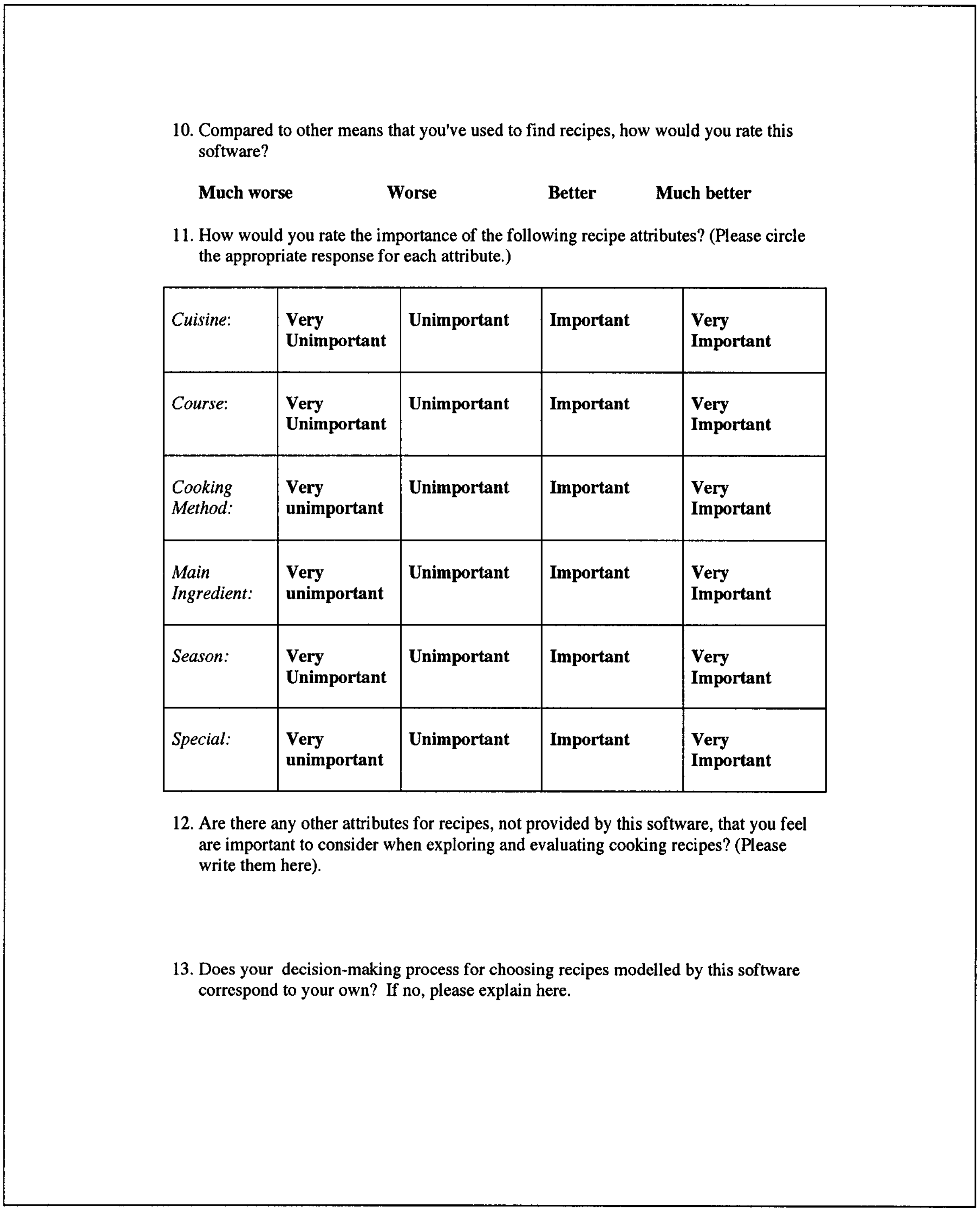
8. How would you rate the usability of the software?

   **Very poor** **Poor** **Good** **Very good**

9. If this software was available to you in your home, would you use it to find recipes?

   **Very unlikely** **Unlikely** **Likely** **Very likely**

Figure B.7: Post-task Questionnaire, page 1



10. Compared to other means that you've used to find recipes, how would you rate this software?

**Much worse** **Worse** **Better** **Much better**

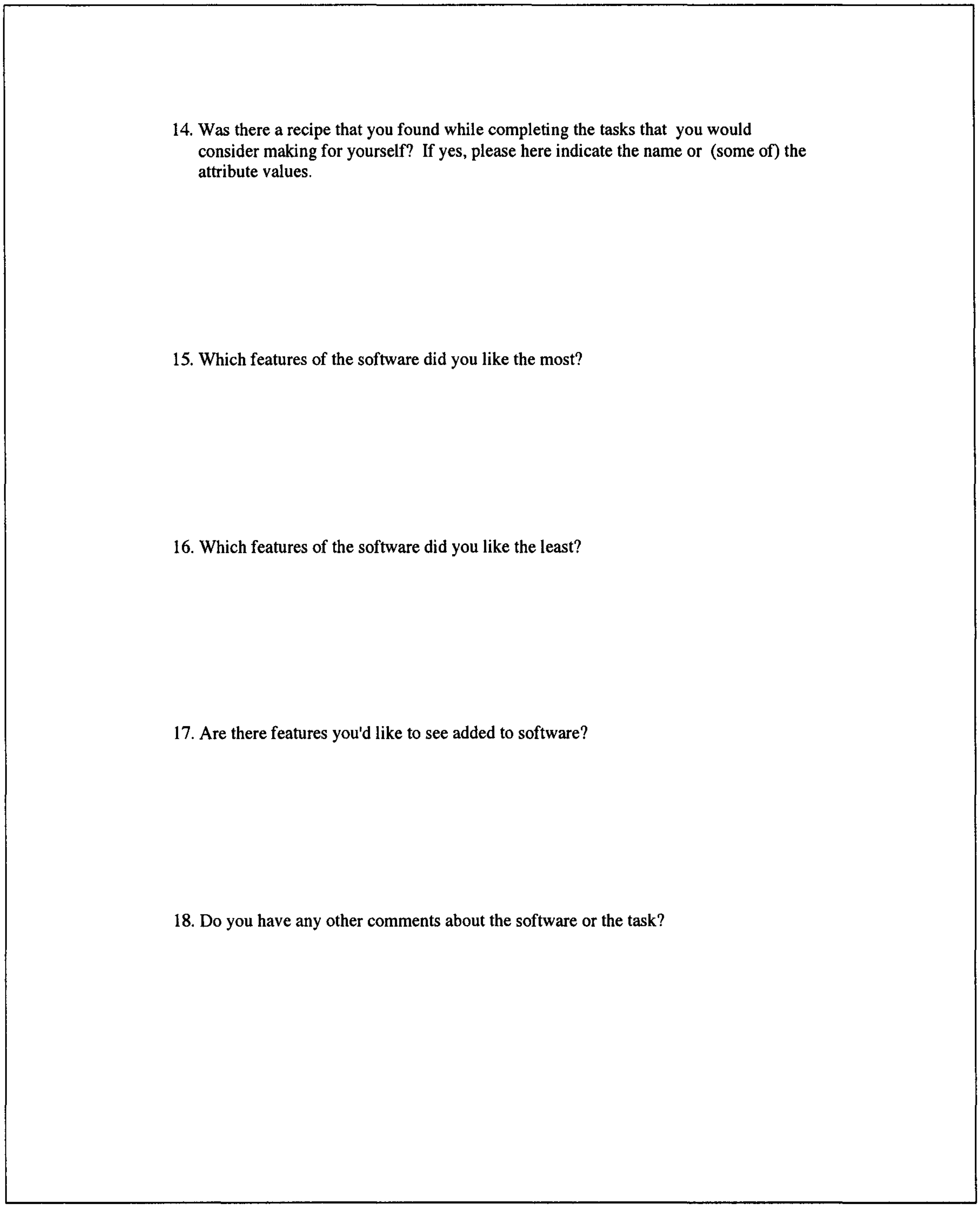
11. How would you rate the importance of the following recipe attributes? (Please circle the appropriate response for each attribute.)

| | | | | |
|---|---|---|---|---|
| *Cuisine*: | **Very Unimportant** | **Unimportant** | **Important** | **Very Important** |
| *Course*: | **Very Unimportant** | **Unimportant** | **Important** | **Very Important** |
| *Cooking Method:* | **Very unimportant** | **Unimportant** | **Important** | **Very Important** |
| *Main Ingredient:* | **Very unimportant** | **Unimportant** | **Important** | **Very Important** |
| *Season:* | **Very Unimportant** | **Unimportant** | **Important** | **Very Important** |
| *Special:* | **Very unimportant** | **Unimportant** | **Important** | **Very Important** |

12. Are there any other attributes for recipes, not provided by this software, that you feel are important to consider when exploring and evaluating cooking recipes? (Please write them here).

13. Does your decision-making process for choosing recipes modelled by this software correspond to your own? If no, please explain here.

Figure B.8: Post-task Questionnaire, page 2



14. Was there a recipe that you found while completing the tasks that you would consider making for yourself? If yes, please here indicate the name or (some of) the attribute values.

15. Which features of the software did you like the most?

16. Which features of the software did you like the least?

17. Are there features you'd like to see added to software?

18. Do you have any other comments about the software or the task?

Figure B.9: Post-task Questionnaire, page 3

# Appendix C

# Selected Recipes

A variety of recipes were chosen by participants and this appendix details the selections made for each task question. Responses which appear in bold were selected by more than one participant.

1. Choose a recipe with American cuisine that you might like to eat.

    - Mashed potato timbales (209, American, Appetizer, Potatoes, Bake)
    - Christmas lane cake (102695, American, Dessert, Holiday, Eggs, Bake)
    - **Sesame chicken wings (204, American, Appetizer, Chicken, Bake)**
    - Parmesan cheese twists (214, American, Bread, Cheese, Bake)
    - **Party rolls (201, American, Appetizer, Beef, Cook)**
    - Almond sunshine citrus (207, American, Appetizer, Fruits, No Cook)
    - Herbed chicken quarters (237, American, Main Course, Chicken, Grilled)
    - Marinated beef pot roast (233, American, Main Course, Beef, Roast)
    - **Scallops on skewers with carrot sauce (206, American, Appetizer, Fish, Grilled)**
    - Mediterranean crescent pinwheels (202, American, Appetizer, Grains, Bake)
    - Melanie's garden-tomato soup (252, American, Soup, Vegetables, Cook)
    - Swiss cheese, spinach and bacon appetizer tarts (203, American, Appetizer, Cheese, Bake)

- Tuxedo cheesecake (224, American, Dessert, Cheese, Bake)
- Peanut butter oatmeal cookies (227, American, Dessert, Eggs, Bake)

2. Find a recipe with a fried cook method and a main ingredient of eggs.

   - Roasted red pepper and zucchini frittata (140, Italian, main course, eggs, cook)
   - **Fried eggs with vegetable confetti (259, American, Brunch, Spring, Eggs, Fried)**
   - **Classic French toast (7410, French, Breakfast, Eggs, Fried)**
   - Tandoori-spiced chicken breasts (101511, Indian, Main Course, Spring, Chicken, Fried)
   - Breakfast stack (261, American, brunch, eggs, fried)
   - Spinach devilled eggs (205, American, Appetizer, Eggs, Quick)
   - fried eggs and asparagus with parmesan (260, American, Brunch, Spring, Eggs, Fried)
   - fried ravioli (110, Italian, Appetizer, Cheese, Fried)

3. Find a recipe with a main ingredient of beans, is intended for the summer season, and has a sauté cooking method.

   - **Succotash (103725, American, Side Dish, Summer, Beans, Saute)**
   - Chicken and white bean chili (107492, American, Main Course, Winter, Beans, Saute)
   - Pork stir-fry with green beans and peanuts (108451, Asian, Main Course, Summer, Beans, Stir-fry)
   - **Summer tomato and Basil Spaghettini (108432, Italian, Main Course, Summer, Vegetables, Saute)**
   - Tuna and white bean salad (108500, American, main course, beans, summer, no cook)

- Three way garlic pasta with beans and peppers (106271, Italian, Main Course, Spring, Beans, Saute)
- Curried tofu with spinach and tomatoes (229, American, main course, beans,saute)
- Barbecued ribs with corn and black-eyed-pea salad (106958, American, Main Course, Summer, Beans, Quick)
- Cavatappi with white beans and golden onions (109735, American, Main Course, Summer, Beans, Saute)
- Sauteed dkirt steak (103668, American, Main Course, Summer, Beef, Saute)
- spice-rubbed chicken breasts with lemon-shallot sauce (106642, American, Main Course, Summer, Chicken, Saute)

4. Imagine you would like to have eggs for breakfast. Choose a recipe that you might like to eat.

   - Fried eggs on corn tortillas with two salads(103387, Mexican, Breakfast, Eggs, Fried)
   - asparagus omelet (101617, French, Breakfast, Eggs, Saute)
   - Chorizo scrambled eggs (105908, Spanish, Breakfast, Winter, Eggs, Cook)
   - Potato pancakes with apple sauce (263, American, Breakfast, eggs, cook)
   - **Eggs florentine (85, French, Breakfast, Eggs, Advance)**
   - **Quick eggs benedict (265,American, Breakfast, Eggs, Quick)**
   - Breakfast tortilla wrap (268, Breakfast, Eggs, Quick)
   - Easy breakfast nacho bake (6454, Mexican, Breakfast, Eggs, Bake)
   - egg stuffed breakfast tomatoes (267, French, Breakfast, Eggs, Bake)
   - Spinach & mushroom frittata (124, Italian, brunch, vegetables, cook)

5. Imagine you are cooking for a group of people and for the appetizer course, you would like to serve something with French cuisine. Choose one of these recipes that you might like to serve.

- Chicken liver terrine (65, French, Appetizer, Chicken, Bake)
- **Shiitake mushroom appetizer (68, French, Appetizer, Vegetables, Quick)**
- Artichokes in olive oil (63, French, Appetizer, Vegetables, Cook)
- **Scallops with mushroom in white wire sauce (107582, French, Appetizer, Winter, Vegetables, Bake)**
- **Beef roulades (64, French, Appetizer, Beef, Cook)**
- Smoked salmon Rillettes (81, French, Appetizer, Fish, Cook)
- Toie gras toasts with greens and verjus port glaze(109098, French, Appetizer, Winter, Vegetables, Bake)

6. Use the embedded web search capability to find recipes that are similar to the recipe you just chose (appetizer course and French cuisine). Choose one of the recipes returned from your web search. In addition to its name, please write down its source (for example, its URL or the name of the web site).

   Recipe names:

   - **Mussels with Vegetables and chardonnay cream**
   - French Bread Appetizer
   - Shrimp spread
   - Appetizer pinata meat balls
   - Mouse au chocolat
   - Hors d'Asparagus
   - Broccoli/cauliflower casserole
   - Appetizer kabobi
   - Shrimp cherchi
   - Bacon, cheese and olive melt
   - French Zucchini Fritters

- Jamaican Grilled Fish
- Blue cheese bites
- Overnight French Toast
- Champagne chicken
- Barras "Goey choco"
- Raisin-apricot glazed ham

Sources:

- http://frenchfood.about.com/od/firstcourses/r/musselschardon.htm
- http://frenchfood.about.com/library/blappeizer.htm
- http://www.cooks.com/rec/search/0.1..00,french_appetizers,FF.html
- http://www.cooks.com/rec/search/0.1..00,winter_vegetable_casserole,FF.html
- http://www.cooks.com/rec/ch/appetizers.html
- http://www.recipezaar.com/r/262/107/92/103/81
- http://www.recipezaar.com/r/118/153
- http://www.recipezaar.com/r/219/81
- http://recipes.timerecordnews.com/results.cfm?cot=8
- http://www.jamaicans.com/cooking/
- http://www.ffcook.com/pages/dbdtin-p.htm
- http://recipes.timesrecordnews.com/results.cfm?cat=8
- http://www.cookierecipes.com

7. Imagine you want a recipe for the winter season. Choose at least 2 recipes that look interesting. Refine your query based on these selections and choose a recipe from the results.

   - Philippine-style chicken adobo (107410, Asia, main course, chicken, winter, bake)

- Chocolate-peppermint ice cream cake (108968, American, Dessert, Winter, Chocolate, Advance)
- Chocolate hazelnut ginger biscotti (102709, Italian, Dessert, Winter, Chocolate, Bake)
- Spaghetti pie with broccoli rabe (104555, Italian, Main course, Vegetables, winter, bake)
- Braised pork with orange and fennel (109014, American, main course, vegetables, winter, quick)
- Warm chocolate tortes with raspberry sauce (127, Italian, dessert, chocolate, bake)
- Sesame wonton crisps (107593, Asian, main course, pork, winter, advance)
- Breakfast sausage casserole (219, American, breakfast, pork, bake)
- Chocolate almond torte (106496, Italian, dessert, chocolate, spring, bake)
- Dulce de leche cheesecake squares (108904, Mexican, dessert, cheese, winter, bake)
- Banana orange crepes (109044, French, breakfast, fruits, winter, quick)
- Hundred corner shrimp balls (103046, Asian, Hors, Shrimp, winter, bake)
- Mango ice cream (29, Asian, Dessert, Fruits, No cook)
- Mango fool (106084, African, Dessert, Fruits, Winter, Cook)
- Spice-crusted rack of lamb (102863, Indian, main course, lamb, winter, roast)
- Greek-style vegetable kebabs with orzo and feta (103621, Greek, main course, vegetables, summer, grilled)
- Polvorones (70, French, bread, grains, bake)
- Jerk pork on red pepper mayo and black-eyed-pen cakes (102728, American, Appetizer, Pork, Winter, Advance)
- Dry curried beans (102946, Indian, Side dish, Beans, Winter, Cook)

# Appendix D

# User Study Comments

This appendix presents the participant responses, in full, to the open-ended questions of the post-task questionnaire. Responses are organized by question.

Q13. Does your decision-making process for choosing recipes modeled by this software correspond to your own? If no, please explain here.

- When I choose recipes, I tend to narrow it down a little (by ingredient say) and browse through recipes then. I don't tend to refine until I find something, but maybe that's because I can't really do that with traditional methods (visual searching is good for me too)
- Yes, when I am going to make dinner I like to know the amount of ingredients I need and the recipes provide this in a clear and logical manner. Also, from the recipes you can see what the health benefits are from this
- Yes, the software was helpful in helping me choose the recipes I want.
- Very close
- I have experience when cooking, so I look at how the recipe is made and then alter it to suit my ingredients. So, if I were cooking pork, I would find pork recipes, combine a few , make my food
- Yes, In some ways. I would use this software to help me find interesting and possible recipes for the ingredients that I currently have, It is also very

useful to be able to find dishes similar to ones you like, I do that when I have the freedom to.

- Yes, It did a reasonably good job of guiding me through the selection process
- Generally looking for a specific food type ingredient (i.e. recipe for chicken or beef). Looking for main stream ingredients. The software covers that.
- Region choosing is good, however due to no support for other languages a lot of recipes would be missing. Also "rustic" recipes would be a nice addition
- I would search by the ingredients I have available, so yes the software helped with that
- No, I am used to flipping through a book when choosing a recipe. I look at the main ingredient category (ie. Chicken) and find all the recipes listed. I scan the ingredient lists, the time it takes to prepare, how many servings, to see decide if I want to make it.

Q14. Was there a recipe you found while completing the tasks that you would consider making for yourself? If yes, please here indicate the name or (some of) the attribute values.

- No.
- I am not sure of the name, but I think it had manager in it. The attributes to find it I think were Season winter.
- I am bad at cooking, so no.
- I did not like to cook that much.
- There were some that looked very good, but I doubt I would have the knowledge or cooking skills to make some of the dishes.
- No, because I don't cook very much.
- Many of them looked interesting - I would try a lot if I had the time.

- Yes. Curried tofu with spinach and tomatoes.
- Not really.
- Dulce de leche cheesecake squares(Mexican, dessert)
- Yes, the attributes were dessert and Chocolate
- Chocolate almond torte; or Summer tomato & basil spaghetti
- Perhaps, though I am not a very good cook. Some of the American cuisine looked very good, eps. Appetizers. Also some shrimp dishes looked wonderful.
- Yes, Sesame seed wing.
- Vegetarian spring rolls.
- Yes, I use all recipes I find. Its an art form that you pick tidbits of information from over a life time. The braised pork looked interesting.
- There were many I would like to make, for example: many of the parches (Almond sunshine citris, tandoori-spiced chicken).
- Other than the Scrambled eggs. No, I can't cook very well.
- Chocolate peppermint ice cream cake.
- There were plenty. As for as attributes go, I like many type of ingredients, so there is hot one particular set that interested me. It all depends on the day and my need, and the food in the house.

Q15. Which features of the software did you like the most?

- To see the recipe visually with a title, the ability to restrict into broad categories.
- The new dialog box
- Large clickable buttons to select recipes, pictures
- The method option The new button that allows you to select what preference you are looking for.

- The pictures of the recipe, ability to see the recipe without losing the search. Full recipe easy to access.
- The pictures really helped with choosing recipes because I did not know what many of the recipes were just by name
- I liked being able to see the food before looking at the recipe
- I enjoyed the visual representation of the food and the searching with selected options ability.
- Attribute values. Let me find exactly what I wanted very quickly. Realistically, a non-computer person would have a hard time with the interface, but to me it is normal.
- Pictures (2)
- The picture showing the dish is good, the search tool (create new space) seems to be fast and effective.
- The wide variety available that can be easily narrowed.
- Narrowing the search to main ingredients and course
- Search function, it was easy to use and quick.
- I like the search refinement tool, however the word "space" and "new" did not really indicate what this really was.
- Being able to go back on query, the way you can decide what cuisine/dish, etc. you want.
- I like the various options like cuisine, ingredient, etc. The pictures of the finished dish was also helpful when choosing.
- I liked the fact that you were able to choose what cuisine, season, etc. and from what culture it was from. I also like the pictures of what you have chosen and having the ability to do a web search and check the recipe for it.

Q16. Which features of the software did you like the least?

- Web search , the html didn't render properly and I like to see the list of results first, then choose the recipe , rather than just go to the first page.
- Web search
- View key
- The view key was kind of confusing
- Cuisine
- Some of the labeling is unclear. You may want to change "space" to "search for recipes" , etc. It is unclear what the view key does.
- The user interface was awkward. The new space dialogue could be better via pull down menus.
- Tiny "new" button, not an obvious meaning of all. When flipping pages, it did not visually keep selection. The web search had trouble with the window size (or maybe source website did) , web search was difficult to understand and use.
- It is slightly unfamiliar, so it took a moment to learn, but once learned, it was easy.
- Not all web links worked and some were slow
- The webpage search returned too much links and some were not relevant.
- Space
- I am not sure about what I don't like. But the "space" box is a bit confusing to me, if I search on my own without any help.
- Not enough recipe options on each page.
- Labels, web search was not very useful as I would just resort to using my own web browser to search google.
- Everything looks good to me.

Q17. Are there features you'd like to see added to software?

- The only thing I can think of is a Not function for selection a recipe without chicken, if I was allergic , for instance.
- Print out recipe in a compact form.
- To search by ingredients
- Bigger "new" button (and different name), more ingredients, maybe a special "spices" area.
- Just being able to search by amount of ingredients you have
- Not unless the software can cook the food, too
- That summary of ingredients on main page.
- An option to store some of your favorite recipes so that you can find them faster next time.
- Language support
- You could show more recipes per page, maybe 12
- Being able to see entire list without pictures or anything
- There are already a lot of options to choose from
- A link that says "would you like to store this search information", something like that and when you click on it a window popup with fields to fill in the information to store
- Better and more clear labeling

Q18. Do you have any other comments about the software or the task?

- The visual searching of recipes is handy compared to a text search
- It is a great idea
- I like this software. Great interface . One of the best cooking recipe applications I have seen
- I would definitely use this if I were in a hurry

- I like the idea of this software a lot. It would help me getting rid of leffover ingredients. I think the layout is good, and once learned, very fast, powerful and easy to use.
- I think the software is excellent, I am very hungry now.
- Some web pages did not display correctly, may want to see if that is the fault of the webpage or some flaw or bug in the program.
- Nice software, almost made me want to start cooking myself.
- Software is easy to use and fast, Good criteria for searching recipes.
- Good idea
- Add more common everyday food like good old mac and cheese etc. Most of the food seems too fancy for something I would make.
- I wasn't sure of what the dropdown list was for the web search(the one with urls) . I though it was a recently used list like in web browsers.
- I would just like to say that this software is great and I hope to see it on the web someday. I know of no other software that does what this software can do.
- There seemed to be a lack of screening for "technical" terms both in the task instructions and the software itself. I found these disorienting and confusing. Layout was appealing, nice use of thumbnail pictures, perhaps ratings or link to review would make this more appealing than a common cookbook.