CASED BASED APPROACH TO SEMANTIC WEB

A Thesis
Presented to the Institute of Science and Engineering
of Işık University
In Partial Fulfillment of the Requirements for the Degree of
Master of Science
in
The Department of Computer Engineering

By
Pınar Demirel
June 2005

ABSTRACT

CASED BASED APPROACH TO SEMANTIC WEB

Demirel, Pınar

World Wide Web (WWW, Web) presents an incredibly rich medium for information sharing and communication. Most, if not all, of the applications utilizing WWW transforms information through HyperText Markup Language (HTML) in order to present it in a form that is understandable by human beings. However, this transformed information targeting humans is usually not easily understood and processed by machines and this inhibits machine access to information residing in WWW and re-usability of this information.

The most popular approach to this problem is based on developing Web applications that are accesible by machines through the usage of Web onthology languages. These Web ontology languages, based on Extensible Markup Language (XML), are used in order to insert onthology data into Web pages. These Web pages containing onthology data can then be accessed and used by those applications (and consequently machines) that have been developed through the same onthology.

Ontology based solutions present various shortcomings and complexities due to the number of different existing Web ontology languages as well as their complex structures.

Case based approach is presented in this paper as an alternative to onthology based solutions. Within the case based approach, information is described as "cases" which are then linked with each other through "special cases". Moreover, "special cases" can also be utilized in order to describe the correlation between "cases" in different domains. These characteristics enable the case based approach to function as more than a simple keyword matching system and as an intelligent system that processes information according to pre-defined rules. As a result, information can be easily accessed and understood by machines for further processing or re-use.

Moreover, case based approach can support distributed systems through its support for correlating information across different domains.

Key Words: WWW, Web, semantic web, cased based

ÖZET

DURUM YAKLAŞIMLI ANLAMLI
WEB UYGULAMASI

Demirel, Pınar

World Wide Web (WWW, Web), bilgi paylaşımı ve iletişim için çok geniş bir ortam sunmaktadır. WWW'deki bilgiye erişmek için kullanılan uygulamaların çoğu bilgiyi insanların anlayabileceği bir şekile çevirmek için HyperText Markup Language (HTML) kullanmaktadır. Fakat insanların anlayabileceği bir şekile çevrilmiş olan bilgi makinalar tarafından kolaylıkla anlaşılamamakta ve bu da WWW'deki bilginin makinalar tarafından erişimini ve yeniden kullanımını kısıtlamaktadır.

Bu soruna çözüm olarak öngörülen en popüler yaklaşım, ontology web dilleri kullanılarak makinaların anlayabileceği WWW uygulamaları geliştirmektir. Bilginin makinaların anlayabileceği bir şekile sokulması için geliştirilen XML tabanlı ontoloji dilleri, ontoloji bilgisinin Web sayfalarına eklenmesi için kullanılmaktadır. Ontoloji bilgisine sahip Web sayfaları aynı ontoloji ile yazılmış programlar tarafından erişilebilir ve Web sayfasındaki bilgiler makinalar tarafından kullanılabilir.

Ontology tabanlı Web çözümleri, geliştirilmiş farklı ontoloji dilleri ve bu dillerin karmaşık yapısı dolayısıyla bazı eksiklikleri ve zorlukları olan bir yaklaşımdır.

Ontoloji tabanlı çözümlere alternatif olarak durum tabanlı yaklaşım önerilmiştir. Bu yaklaşımda bilgiler durumlar olarak tasvir edilir ve durumlar arasındaki ilişkileri belirten ve farklı alanlardaki durumları ilişkilendiren özel durumlar yaratılabilir. Bu özelliğiyle durum yaklaşımlı sorgula sistemi, sadece anahtar kelime eşlestirmesi yapan bir sistem yerine, bilgiyi tanımlanan kurallara gore işleyen akıllı bir sistem haline getirilir. Böylece, kullanıcının hedeflediği bilgi, daha kolay anlaşılır ve ulaşılabilinir. Durum tabanlı yaklaşım farklı alanlardaki durumları da ilişkilendiren yapısıyla, dağınık sistemleri de destekleyebilir.

Anahtar Kelimeler: WWW, Web, anlamlı web, durum yaklaşımı

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Prof. Dr. Selahattin Kuru, for his guidance, patience and support throughout this thesis. I also thank to my family for their support and motivation.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

The creation of World Wide Web (WWW, Web) has introduced an incredibly rich information-sharing environment into our lives. Since the introduction of WWW, lots of standards have been implemented for information presentation, sharing, and retrieval through the Web and mainly for human interpretation and consumption. The current Web uses HyperText Markup Language (HTML) to display content and it provides a uniform syntax in which publishers can present information in a human-readable form through the usage of Web browsers.

Although WWW contains very large amount of information, user cannot fully benefit from it because of limitations imposed by current standards. With the current standards, finding information on the Web is based on retrieving information through matching keywords, or specific tags within a document. Users need to manually find the relevant information they are looking for in search results which very often requires reasoning exhibited by humans but very hard to emulate by machines. As a result, information presented by the current Web is not precise enough for machines to 'understand' and process.

To make the Web more useful, information content should be processed meaningfully and this is where semantic web is called into action. The idea behind semantic web is to encode structure and semantics into Web documents so that machines can meaningfully process its contents.

Ontology based approach is the most popular method used for semantic web implementations. However, the complex structure of the ontology approach brings bottlenecks and deficiencies to these semantic web implementations.

In this thesis, we propose Case Based systems as a complement and alternative to Ontology based systems in order to offer an innovative, context-aware and intelligent approach to semantic web. The Case Based system represents information and experience as "cases".

The thesis will introduce the case based approach, compare it with the ontology approach, propose a case based implementation model for semantic web and provide a sample case base implementation very innovatively named "Case Based Semantic Web".

2. CASE BASED APPROACH

Case-based reasoning (CBR) was first formalized in the 1980s following from the work of Roger Schank and his students at Yale University, and is based upon the fundamental premise that similar problems are best solved with similar solutions. The idea is to learn from experience. CBR tackles arising problems based on the solutions of similar problems that have been tackled in the past. For example, in very simple terms, a mechanic who fixes an engine based on his experience with another engine that had exhibited similar symptoms as this one is using case-based reasoning. A lawyer who advocates a particular outcome in a trial based on judgments given on similar trials in the past or a judge who creates case law is using case-based reasoning.

CBR is a technique for managing and using knowledge that can be organized as discrete abstractions of events or entities that are limited in time and space. Each such abstraction is termed a case. Software engineering examples could be projects, design patterns or software components. Cases are characterized by vectors of features such as file size, number of interfaces or development method. CBR systems typically function by solving the new problem, often termed the target case, through retrieving and then adapting similar cases from a repository of past (and therefore solved) cases. The repository is termed the case-base.
CBR is argued to offer a number of advantages over many other knowledge management techniques, in that it:

- avoids many of problems associated with knowledge elicitation and codification.
- only needs to address those problems that actually occur, whilst generative (i.e. algorithmic) systems must handle all possible problems.
- handles failed cases, which enable users to identify potentially high risk situations.
- copes with poorly understood domains (for example, the web is such a domain) since solutions are based upon what has actually happened as opposed to hypothesized models.

- supports better collaboration with users who are often more willing to accept solutions from analogy based systems since these are derived from a form of reasoning akin to human problem solving. This final advantage is particularly important if systems are not only to be deployed, but also to have trust placed in them.

It can be argued that case-based reasoning is not only a powerful method for machine reasoning, but also an important aspect of behavior in everyday human problem solving and reasoning. Moreover, a stronger statement that all human reasoning is based on past experiences and is accepted inevitably by every active choice being made is supported by a part of the scientific community interested in human cognitive science.

At case based semantic web, we combine case based reasoning and knowledge representation approaches, to make system learn from previous experiences.

2.1 Objectives of Case Based Approach in Semantic Web

Experience and Knowledge Representation: In essence, we represent knowledge, experience and the relationship between knowledge as cases. So the very first problem arises in the representation model of cases. Basically, we represent an individual case in a case base as (Definition of the Problem in case base terminology, Definition of the Solution in case base terminology, Definition of the Outcome in case base terminology) [1]. Cases, which comprise problems and their solutions, can be used to derive solutions to new problems. However cases comprising problems and outcomes can be used to evaluate new situations. If, in addition, such cases contain solutions they can be used to evaluate the outcome of proposed solutions and prevent potential problems.

There is a lack of consensus within the case based research community as to exactly what information should be in a case. However, two pragmatic measures can be taken into account in deciding what should be represented in cases: the functionality and the ease of acquisition of the information represented in the case.

Indexing: Case indexing involves assigning indices to cases to facilitate their retrieval [2]. Indices should:

- be predictive,
- address the purposes the case will be used for,
- be abstract enough to allow for widening the future use of the case-base, and
- be concrete enough to be recognized in future.

Both manual and automated methods have been used to select indices. Choosing indices manually involves deciding a case's purpose with respect to the aims of the reasoner and deciding under what circumstances the case will be useful. Other methods for indexing are:

- Indexing cases by features and dimensions that tend to be predictive across the entire domain i.e., descriptors of the case which are responsible for solving it or which influence its outcome. In this method the domain is analyzed and the dimensions that tend to be important are computed. These are put in a checklist and all cases are indexed by their values along these dimensions.

- Difference-based indexing selects indices that differentiate a case from other cases. During this process the system discovers which features of a case differentiate it from other similar cases, choosing as indices those features that differentiate cases best.

- Similarity and explanation-based generalization methods, which produce an appropriate set of indices for abstract cases created from cases that share some common set of features, whilst the unshared features are used as indices to the original cases.

- Inductive learning methods, which identify predictive features that are then used as indices.

Explanation-based techniques, determines relevant features for each case. This method analyses each case to find which of their features predictive ones are. Then, cases are indexed by those features.

Storage: Case storage is an important aspect in designing efficient case base systems in that, it should reflect the conceptual view of what is represented in the case and take into account the indices that characterize the case. The case-base should be organized into a manageable structure that supports efficient search and retrieval methods. A balance has to be found between storing methods that preserve the semantic richness of cases and their indices and methods that simplify the access and retrieval of relevant cases. These methods are usually referred to as case memory models. The two most influential case memory models are the dynamic memory model where the case memory model in this method is comprised of memory organization packets or MOPs [3], and the category-exemplar model where the model organizes cases based on the view that the real world should be defined extensionally with cases being referred to as exemplars.

Information Retrieval: Given a description of a problem, a retrieval algorithm, using the indices in the case-memory, should retrieve the most similar cases to the current problem or situation. The retrieval algorithm relies on the indices and the organization of the memory to direct the search to potentially useful cases.

The issue of choosing the best matching case has been addressed by research into analogy [4]. This approach involves using heuristics to constrain and direct the search. Several algorithms have been implemented to retrieve appropriate cases, for example: serial search, hierarchical search and simulated parallel search.

Case-based approach will be ready for large-scale problems only when retrieval algorithms are efficient at handling thousands of cases. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case.

Among well known methods for case retrieval are: nearest neighbor which involves the assessment of similarity between stored cases and the new input case, based on matching a weighted sum of features; induction where algorithms determine which features do the best job in discriminating cases, and generate a decision tree type structure to organize the cases in memory; knowledge guided induction and template retrieval. These methods can be used alone or combined into hybrid retrieval strategies.

Adaptation: Once a matching case is retrieved a case-based system should adapt the solution stored in the retrieved case to the needs of the current case [5]. Adaptation looks for prominent differences between the retrieved case and the current case and then applies formulae or rules that take those differences into account when suggesting a solution. In general, there are two kinds of adaptation in case-based: Structural adaptation and derivational adaptation. Adaptability of cases will also facilitate a context-aware case base system that can adapt itself according to the context and user-needs.

Interoperability and Sharing: Together with the case base, a case-based approach for sharing knowledge should be defined. In many research conducted, XML is taken as a foundation for case-based sharing. A number of XML based mark-up languages have been developed but none are fit to be standards. The Case Base Mark-up Language (CBML) should be at the place where case-based approach meets semantic web.

To make traditional case languages adapted to the usage in the semantic web, namely, facilitating the encoding of case knowledge into web documents, the following issues should be taken into consideration:

- Some standard vocabularies for case description are needed, which ensure the success of case interchanging and distributed case-based approach.

- Some conveniences for integrating domain vocabularies should be provided. Domain vocabularies provide the means of defining classes, attributes, and types.

- The web case language should be flexible so that it could fulfill the needs of both unstructured and structured case representations.

The CBML facilitates semantic interoperability between systems, allowing definition of all types of case bases in various digital forms.

Case Modification: The case base management system should also address the modification of the stored cases and allow basic arithmetical operation to be executed on the case base, like case addition, subtraction, division, etc. Furthermore, relationships between cases should be modified for adaptation. Relationships like, ‘is-a’, ‘part-of’, ‘includes’ should be developed to facilitate maintainability and extensibility of the case-base.

Case Based Knowledge Acquisition from Various Resources: In addition to the case base management system, some authority tools should be provided from outside knowledge acquisition. Knowledge from resources like, FAQ’s, forum, or newsgroups, etc. are to be represented in the case base approach and should be added and adapted to the case base. Even multimedia content may be represented as cases, with a more experience based, data-driven approach. Thus, here we address creation of tools that can do automated knowledge discovery, acquisition and sharing of various types of digital content.

2.2 Other Systems That Use Case Based Approach

Cases have been used extensively for professional preparation in law, medicine, and business, but are only now beginning to be widely employed in education [6]. Cases can involve a description of real events, as is common in business cases that describe the processes at work in an actual corporation, or they can present a fiction based upon real-life but constructed to provide specific experiences for students. Instructional cases can be used to encourage the development of professional thinking as individuals formulate reactions to case materials. Case methodology is especially effective if students are required to identify facts and issues, to view events from different perspectives, to apply current professional knowledge and research, and to predict consequences of various courses of action [7]. In this way, the use of case methods can help students to forge important connections between the academic and the experiential, between knowledge and practice [7]. The effectiveness of case-based teaching is supported by Kleinfeld [8], who has demonstrated that teaching with cases can help students to understand the meaning of events, increase their ability to frame educational problems, and improve their thinking regarding alternative courses of action.

Cases are like stories or story lines that students read or explore interactively. They can direct students toward a conclusion, or provide the resources and context to discuss and debate issues dynamically.

Case-based learning becomes case-based reasoning when more than one case is provided. Case-based reasoning involves reasoning about multiple cases and how prior solutions can be adapted to new problems or how prior cases are related to new cases.

Cases are best used to teach people about realistic decision-making situations. For instance, cases help train pre-service teachers, instructional designers, doctors, lawyers, business people, and others, how to respond to actual problems they will encounter in their fields. Cases are used primarily for corporate and industrial training and across many professional university programs.

Case-based learning is promoted in many universities because it teaches important concepts and facts within the context of authentic or real-world situations. Context is thought to be more motivational to learners and it provides a concrete framework from which complex concepts can be more easily understood.

Further, case-based learning reduces the potential for “inert” knowledge. Inert knowledge is learned information that is difficult or impossible to apply to realistic situations, because it was learned in a “chunked” fashion, typically out of all context or relation to reality. When learning statistics, for example, simply crunching numbers may leave a student with “inert” knowledge of formulas. If taught in the context of a research design relevant to the student’s field, however, the familiar context may help the student understand the relevance of that information. Isolated facts are more difficult to integrate within memories than facts taught in a realistic context.

The method of case-based learning is a learner-centered approach, which includes students’ research and interactions with each other while studying an object or problem of special importance. After having developed a case, the teacher usually makes an introduction to a subject and presents a case to students. The students examine the objects from different perspectives, collect and analyze data pertaining to the case, and discuss feasible ways of solving the case. The teacher can help them in any activity.

A recent research on pedagogical issues and models for e-learning with case based learning approach has some findings about building some pedagogical designs for digital learning systems [9]. Ip and Naidu (2001) made a selection of experience-based pedagogical designs, which stand to make the most of the opportunities afforded by information and communications technology, grouped the use of experience in pedagogical designs into first-person experience-based design and third-person experienced-based design. First-hand-experience-based designs are learning environments that provide a safe and authentic environment for learners for thinking, reflection, decision-making, making mistakes and learning from their own experience. Learning environments with first-person experience-based design include role-play simulation and rule-based simulation. Third-person-experience-based designs are learning environments that make extensive use of real-life stories to support the learning and teaching process. These designs are grounded in the belief that stories and experience comprise not only the most authentic repository of knowledge but also serve as a strong motivator of learning.

3. CASED BASED VERSUS ONTOLOGY BASED SEMANTIC WEB

Ontology based approach is the most popular method for implementing content aware Web systems in order to make the information available in a Web page accessible for machines. A basic introduction to ontology approach and the technologies involved is presented below together with its shortcomings and complexities compared with a case based approach.

3.1 Ontology Based Semantic Web

Semantic web transforms the existing web pages in order to have the information contained in these web pages understandable for machines so that computers can use and further process this information. For example, databases, programs, services, personal devices, etc. can use the information present in the Web to search, filter and prepare useful information for their human users.

To make all this more concrete, real world applications of semantic web include intelligent search agents, data modeling solutions, and e-learning applications that search, filter and present specific sets of data present in the Web.

XML, XML Schema, RDF, RDF Schema and OWL are the tools that are utilized in ontology based semantic web [10]. The OWL Web Ontology Language Overview (http://www.w3.org/TR/owl-features/) describes the function and relationship of each of these tools as follows:

- XML [11] provides an overall syntax and structure for documents but does not further specify any constraints as to the meaning of these documents

- XML Schema is a language for restricting the structure of XML documents

- RDF [12] is a data model for objects and relations between them. It provides simple semantics for these data models that can then be represented in XML syntax,

- RDF Schema presents a vocabulary for describing properties and classes of RDF resources as well as a semantics for building hierarchies of these properties and classes

- OWL presents further vocabulary for describing properties and classes through its ability to define relations between classes (disjointness, cardinality, etc.) and its richness in representing properties as well as characteristics of properties.

The main enablers of the tools mentioned above are URIs [2] which identify resources along with XML and namespaces. These enablers, together with a bit of logic, form RDF that can be used to say anything about anything. Other than RDF, many other technologies such as Topic Maps and pre-web artificial intelligence technologies are likely to contribute to the semantic web.

3.2 Technologies Used At Ontology Based Semantic Web

3.2.1 XML

Semantic web aims to add structure and semantics to web pages so that the information contained in these web pages is accessible by machines for further use. The whole process of semantic web is necessitated by the fact that current Web applications are focused on presenting data to human users through the utilization of HTML. HTML encoded Web pages are difficult to be interpreted by machines. On the other hand, the last few years have experienced a rise in popularity for Extensible Markup Language (XML) that is used for storing and exchanging data between machines. As the name itself indicates, XML is "extensible" meaning that it provides a general structure and data format that should further be enhanced through user defined data tags. XML enforces a strict syntax for documents, but does not imply a specific interpretation of the data [13]. The rising popularity of XML only naturally makes it an ideal candidate for use in semantic web applications as a start for structuring information.

For further clarification of XML and the data tags, let's take the example of the tag "chair." The "chair" tag may simultaneously refer to the object that we sit on or to the head of a company. In these cases, namespaces are used in order to distinguish between different meanings referred to by the same XML tags.

However, XML cannot be used to represent complex information due to its lack of ability to specify semantics. To further enhance XML, specifications like DTD (Document Type Definition) and XML Schema have been developed to describe XML data structures, the names of elements and attributes, and their use in documents [13]. However, DTD and XML Schema are only a step forward from pure XML and not a significant jump since they too cannot specify data meaning within documents.

XML Schema specification also employs name spaces to categorize data. An XML namespace is a collection of names, identified by a URI reference, which are used in XML documents as element types and attribute names [14]. The names defined in an XML schema belong to a target namespace. Declarations and terms in an XML schema can then refer to names belonging to other namespaces.

Below is an example that shows namespace declaration within an XML Schema fragment:

```
<example:schema
        targetNamespace = “http://www.example.com/example1”
        xmlns:example = “http://www.w3.org/1999/XMLSchema”>
<example:element name = “SSN” type = “example:positive-integer” />
```

All elements starting with "example:" are understood as referring to rules and terms defined by "http://www.w3.org/1999/XMLSchema" namespace. In the fragment, the elements ‘schema’, ‘element’, and ‘positive-integer’ belong to the namespace http://www.w3.org/1999/XMLSchema.

As can be seen from the fragment above, XML and XML Schema allows for categorization of data along with predefined rules. These predefined rules enable machines to be able to interpret information contained in XML documents; however, in terms of semantic web applications, this is still far from a complete solution mainly due to lack of any tools for defining data meaning.

3.2.2 RDF

RDF (Resource Description Framework) [15] is a model for representing data about resources on the Web. It is built on top of XML, and represents data in the form of triples – resource, property and statement. For example, the statement ‘The author of this paper is Pinar Yoruk’ has the following triples: ‘Pinar Yoruk’ – statement, ‘author’ – property, and ‘paper’ – resource.

The following is an RDF example, establishes the relationship between document and its author and homepage.

```
<?xml version="1.0"?>
<RDF>
        <Description about="http://www.pamiryoruk.com/documents">
                <author>Pamir Yoruk</author>
                 <homepage> http://www.pamiryoruk.com </homepage>
        </Description>
</RDF>
```

First statement of this rdf example is: "The author of http://www.documentpage.com is Pamir Yoruk".

- The subject of the statement above is: http://www.pamiryoruk.com/documents
- The predicate is: author
- The object is: Pamir Yoruk

Second statement of the RDF example is: "The homepage of http://www.pamiryoruk.com/documents is http://www.pamiryoruk.com".

- The subject of the statement above is: http://www.pamiryoruk.com/documents
- The predicate is: homepage
- The object is: http://www.pamiryoruk.com

The RDF model does not sufficiently define the semantics of the application domain. It just provides a domain-neutral mechanism to describe metadata [13]. RDFS (RDF Schema) is a type system for RDF and it allows the user to define class hierarchies, specify properties, and enforce domain and range specifications for these properties. However, RDF and RDFS are still limited as a knowledge representation language due to their lack of support for advanced features like defining properties of properties (unique, transitive, inverse), disjoint classes, and so forth.

3.2.3 DAML+OIL

Although RDF and RDF Schema allow users to define class hierarchies, specify properties, and enforce domain and range specifications for these properties, they still lacks the necessary expressive power for many applications. DARPA Agent Markup Language (DAML) and the Ontology Inference Layer (OIL), is an attempt to address shortcomings of the RDF and RDFS specification by incorporating additional semantic features. DAML and OIL have been proposed and finally merged into DAML+OIL because of their similarity.

DAML+OIL is based on description logics encoded in RDF. In addition to RDF(S), DAML+OIL provides the ability to express the equivalence or disjointness of classes, additional restrictions like cardinality, or to build new classes as intersections or complements of other classes. Furthermore, DAML has been integrated with XML Schema providing a rich set of data types, which are still missing in RDF(S).

Because of these added semantics, if you tell a computer something in DAML, it can give you new information, based entirely on the DAML standard itself. DAML+OIL gives computers one extra small degree of autonomy that can help them do more useful work for people. For example with following DAML statements: (motherOf subProperty parentOf), if we a triple stated as (Sevim motherOf Pinar), DAML allows you to conclude: (Sevim parentOf Pinar) based on the logical definition of subProperty as given in the DAML specification.

The further evolution of DAML+OIL is the Ontology Web Language(OWL).

3.2.4 OWL

OWL (Ontology Web Language) builds upon what is provided with XML and RDF in order to further provide semantics.

Semantic Strengths of OWL

OWL provide different properties like transitivity, symmetry, function, inverse etc. With transitivity relation user can define a relation like: "Joe siblingOf Sally and Sally siblingOf Bob then Joe siblingOf Bob". With inverse relation user can define a statement like: "Joe siblingOf Sally then Sally siblingOf Joe"

Cardinality restrictions as specifying how many elements are in relation with each other, can also be define at OWL. Cardinality restrictions, contains keyword like: at-most, at-least, exactly, optionality (0 or more). This statement can be given as an example to cardinality restrictions that can be defined at OWL: "Bird has (exactly) 2 Wing".

Type restrictions are the other relations that can be used at OWL. It identifies subclasses that have some restriction on a property P. As an example:

•Bottle madeOf Material

•Glass subclassOf Material

•GlassBottle subclassOf Bottle madeOf Glass

Same concepts can be included in different ontologies. At OWL "equivalence of concepts" allows concepts that have been defined in different antologies to be equated, like:

Ont1:LiquidContainer equvalentClass ont2:Bottle

As a consequence, any instance of LiquidContainer is also an instance of Bottle and any instance of Bottle is also an instance of LiquidContainer. "Equivalence of individuals" allows instances defined in different ontologies to describe the same objects and difference of individuals assert values that are mutually distinct.

OWL also has complex types to support set theory with union, intersection and complement property. As an example this statement can be defined at OWL with these complex types: "Human= unionOf Woman and Man".

Enumerated classes of OWL specify a class via a direct enumeration of its members, this is done through the oneOf construct as an example: “Gender oneOf male or female”. Disjoint Classes of OWL guarantees that an individual that is a member of one class cannot simultaneously be an instance of a specified other class, as an example: “Cat disjoint Dog”.

Inheritance (Subsumption) relation used to inherit the properties of the super class to the subclass and eventually to the instances.

At OWL Automatic classification is use to define a new object of type Thing (the top level node) and depending on its property that object is automatically classified (and completed)

•Bird subclassOf Animal with exactly 2 Wing

•Tweedy is an instance of Thing

•Tweedy has wing LeftWing

•Tweedy has wing RightWing

•Tweedy has not more wings

•Tweedy is automatically classified as Bird

OWL Full, OWL DL, and OWL Lite

OWL consists of three increasingly expressive sublanguages designed for use by specific communities of implementers and users. These sub languages: OWL Lite, OWL DL and OWL Full are described below [10].

- *OWL Lite* supports those applications that mainly need a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1 through a simple true or false logic. Compared with other OWL languages, it enables data to be accessible through much simpler tools with the cost of lack of detailed expression

- *OWL DL* supports those applications that wish the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time). OWL DL includes all OWL language constructs, but they can be used only under certain restrictions (for example, while a class may be a subclass of many classes, a class cannot be an instance of another class). OWL DL is named due to its similarity with description logics, a field of research that has studied the logics that form the formal foundation of OWL.

- *OWL Full* supports those applications that wish the maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support complete reasoning for every feature of OWL Full.

Each of these sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. The following set of relations hold. Their inverses do not.

- Every legal OWL Lite ontology is a legal OWL DL ontology.
- Every legal OWL DL ontology is a legal OWL Full ontology.
- Every valid OWL Lite conclusion is a valid OWL DL conclusion.
- Every valid OWL DL conclusion is a valid OWL Full conclusion.

Ontology developers adopting OWL should consider which sublanguage best suits their needs. The choice between OWL Lite and OWL DL depends on the extent to which users require the more-expressive constructs provided by OWL DL. The choice between OWL DL and OWL Full mainly depends on the extent to which users require the meta-modeling facilities of RDF Schema (e.g. defining classes of classes, or attaching properties to classes). When using OWL Full as compared to OWL DL, reasoning support is less predictable since complete OWL Full implementations do not currently exist.

OWL Full can be viewed as an extension of RDF, while OWL Lite and OWL DL can be viewed as extensions of a restricted view of RDF. Every OWL (Lite, DL, Full) document is an RDF document, and every RDF document is an OWL Full document, but only some RDF documents will be a legal OWL Lite or OWL DL documents. Because of this, some care has to be taken when a user wants to migrate an RDF document to OWL.

Comparison of OWL Full, OWL DL, and OWL Lite

OWL Full has whole capability that has been explained at the "Strengths of the OWL" part. Further, it supports mixing RDF Schema definitions with OWL definitions. The advantage of the Full version of OWL is that you get the full power of the OWL language. The disadvantage of the Full version of OWL is that it is difficult to build a full tool. Also, the user of a full-compliant tool may not get a quick and complete answer.

At OWL DL, owl:cardinality with TransitiveProperty can not be used. Also, you can't use a class as a member of another class, i.e., you cannot have metaclasses. FunctionalProperty and InverseFunctionalProperty cannot be used with datatypes, they can only be used with ObjectProperty.

OWL Lite has all the OWL DL restrictions plus not using owl:minCardinality or owl:maxCardinality. The only allowed values for owl:cardinality is 0 and 1. At OWL Lite you can not use owl:hasValue, owl:disjointWith, owl:one of, owl:complementOf and owl:unionOf.

The advantage of the DL or Lite version of OWL is that, tools can be built more quickly and easily, and users can expect responses from such tools to come quicker and be more complete. The disadvantage of the DL or Lite version of OWL is that you don't have access to the full power of the language.

3.2 Comparison of Two Approaches to Semantic Web

The World Wide Web is a huge resource both in terms of the amount of available information and of the growth rate of human users. The restrictiveness of the Hypertext

Transfer Protocol (HTTP) and the early versions of the Hypertext Mark-up Language (HTML) allowed software developers, information providers and end-users to create easy access of the new media and make it available for public masses. However, this simplicity and restrictiveness hampers further development of the web by creating bottlenecks. This raises the need for the advancement of the technologies that are currently and popularly used. Therefore, what we currently experience is a very early version of the web, and the next version will probably be bigger and more powerful.

The vision of semantic web, is the idea of having data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications [16]. It tries to address problems in searching, extracting, maintaining and generating information located in the web.

The most popular methodology for semantic is the ontology based approach. Ontologies provide consensual and formal domain theories enabling knowledge sharing and reuse [16]. The ontology bases standards enable machine-understandable semantics of information. Millions of small-specialized reasoning services provide support in automated task achievement based on the accessible information. However, the ontology approach brings bottlenecks and deficiencies to the semantic web.

The case based approach aims to complement and even challenge the ontology approach for semantic web.

- *Case Based Approach is Appropriate for Specific Domains:* End users and developers have problems utilizing the ontology approach for some specific domains. The ontology approach largely requires very well-defined and structured domain information. Using ontology approach in these domains result in inefficient models. Case-based approach is most suitable for areas that do not have a strong theoretical domain, where rules and ill-defined, incomplete or inconsistent [16]. The web is such a domain and ontology approach has deficiencies for in this area.

- *Case Based Approach Facilitates Usage of Cross Domain Knowledge:* Utilizing already existing information in some domain to another domain is a controversial issue in ontology based approach. Hence, creation of ontologies for many domains is an obstacle faced by developers and end-users. This is a deficiency in the ontology approach is the ontology interoperability from cross-domain interoperability. The idea of a single and overall ontology looks naïve. The size of the ontology would be rather huge. However, analogy-based methodologies in cased based approach facilitate characterising methods to solve new problems based on past cases from a different domain.

- *Semantic Interoperability is Better Addressed in Case Based Approach:* Deriving from the above issue, semantic interoperability in ontology approach has significant constraints. A good example is electronic commerce. Even though standard ontologies exist for e-commerce, they have important deficiencies, e.g., there's no multiple inheritance nor attributes for products. This is an outcome of the complexity problem of the semantic approach. Case based approach, however, reduces complexity of the knowledge represented by using techniques described further on.

- *Case Based Approach Provides a Higher Level of Abstraction:* As we have discussed above, ontology approach has a significant complexity that creates a bottleneck. Case based approach has a higher level of abstraction in this sense. This higher level of abstraction drastically reduces the level of complexity. Both the developers and the end-users of the semantic web benefits from this lower level of complexity. Another facility of a higher level of abstraction is that the amount of data stored for a particular knowledge is reduced significantly. This helps the case based approach in many different ways, like better performance in querying, less need for storage capacity, etc.

- *Case Based Knowledge Evolves According to User Needs:* Ontology developers generally use only knowledge represented in formal ontologies and knowledge bases. There are, however, many other electronic resources that include ontological information that are not explicit ontologies. For example, a medical expert researching cancer therapies and surgical operations need to use information already available in electronic form but represented in a different from, which raw or any other form including multi media content. When the ontology that the researcher is using is different from the one that should be acquired, the information becomes hard to reach, or even unavailable.

Case based approach has competitive advantages in this area. The approach uses the stored cases in this domain to locate any other information that is readily available in the case base. The methods for searching a case base give a higher rate of relevance for a query. This means that user will be able to locate a specified knowledge faster from a more relevant set of search results. This also eliminates unwanted, unsolicited results, like advertisements, to be excluded from the result set. Furthermore, we must note the case based approach is adaptive one, as described further on.

- *Information Retrieval is Addressed Better in Case Based Approach:* End users search the semantic web that is defined by the developer. The search is conducted on general knowledge mostly by using a logic-based search mechanism. Ontology approach commonly utilizes logic-based search which rigorous in the open-world settings of the world which are confronted by the problem of semantic inconsistency. Ontology based approaches behave well under the closed-world assumption that the choices and constraints of domains are known definitely and fixed. In the open environment, we do not know exactly what and how many knowledge bases would participate in the inference process from the very beginning; it is a dynamic process.

  However, reasoning by reusing similar past cases is a powerful and common way to solve problems for humans. Instead of talking about equality, rigorous proof and truth maintenance, case-based reasoning emphasizes approximation, inexactness, and similarities. Case based process is similarity-based. Querying knowledge is much easier for experience than for general knowledge. Furthermore, capturing and maintaining experience knowledge is much easier than capturing and maintaining general knowledge. Also, Case provides users with the conveniences of stating queries incompletely. In most majorities of cases, we do not know exactly what we want to search, but an approximate description of the problems instead. These are important advantages of an experience based approach like the case based approach.

Stating those challenges and objectives we may give some concrete example for the advantages of the case based approach when compared to the contemporary methods:

One of the challenges is to find appropriate specific use-cases, and adapt the case-based methodology to existing methods of information retrieval and knowledge management.

For example, Lawyers already consult large online corpora of case-histories and judgements, but the process of retrieval is largely manual, with individual skill and experience with that system being the main factor in finding the right information. The key is to bring this almost-intangible "skill and experience" into a model that can be replicated and benefit others who might not be as advanced in the field. This involves sophisticated methods of measuring similarities and dissimilarities. The approach would involve taking one domain problem, solving it with the CBMS, and trying to group other categories of domain problems into it, thereby generalizing the same method for larger sets of problems. This would involve liasioning with other organizations, such as law firms, medical practitioners, and other researchers, and stepping through the problem-solving process with them. One way to attract them would be to offer free or subsidized use of end-user tools in return for their inputs on system usability.

As an another example, suppose a geneticist case, she is using a program for viewing micro array data about mice and notices that in the current data set, a number of genes appear over expressed in one condition. She retrieves a sequence for the gene from a database, and submits that sequence to Case Based system that uses human DNA libraries. From the results she compiles a list of human genes that are associated with similar sequences. These results are then passed to another service that queries a Gene Ontology database for all functions that have been identified for those gene names.

This is a discovery activity that requires a number of specialized services to interact with one another to yield an interesting result. Some of these services may be lookup tables that refer to huge data GRID repositories. Other services will have to resolve the semantic differences between description tags used in records for one resource and tags used to describe records in another resource. In other cases, a scenario may require applying the same SQL query to databases whose schemas are slightly different. More commonly, valuable information may have to be extracted from free format text fields of flat files. These services require additional program logic to resolve problems associated with semantic heterogeneity of life science data sets.

4. PROPOSED CASE BASED APPROACH TO SEMANTIC WEB

At the case based semantic web approach (CSWA), we propose to define cases for making to make web a content aware system. The functionality and the ease of acquisition of the information represented in the case will determine how this knowledge is incorporated in the environment. The system indexes the cases, like conventional database management systems. It stores the cases in a manageable structure that supports efficient search and retrieval methods. At our case study we use MySQL database for storing cases. The retrieval mechanism of the case based semantic web, given a description of the problem, using the indexes and a retrieval algorithm, retrieves the most similar cases to the current problem or situation. Unlike database searches that target a specific value in a record, retrieval of cases from the case-base must be equipped with heuristics that perform partial matches, since in general there is no existing case that exactly matches the new case.

The CSWA also facilitates the modification of cases and allow addition, removal of cases. Furthermore, relationships between cases should be modified for adaptation. Relationships like, 'is-a', 'part-of', 'includes' exist to facilitate maintainability and extensibility of the case-base. Figure 4.1 represents the architecture of proposed CSWA.

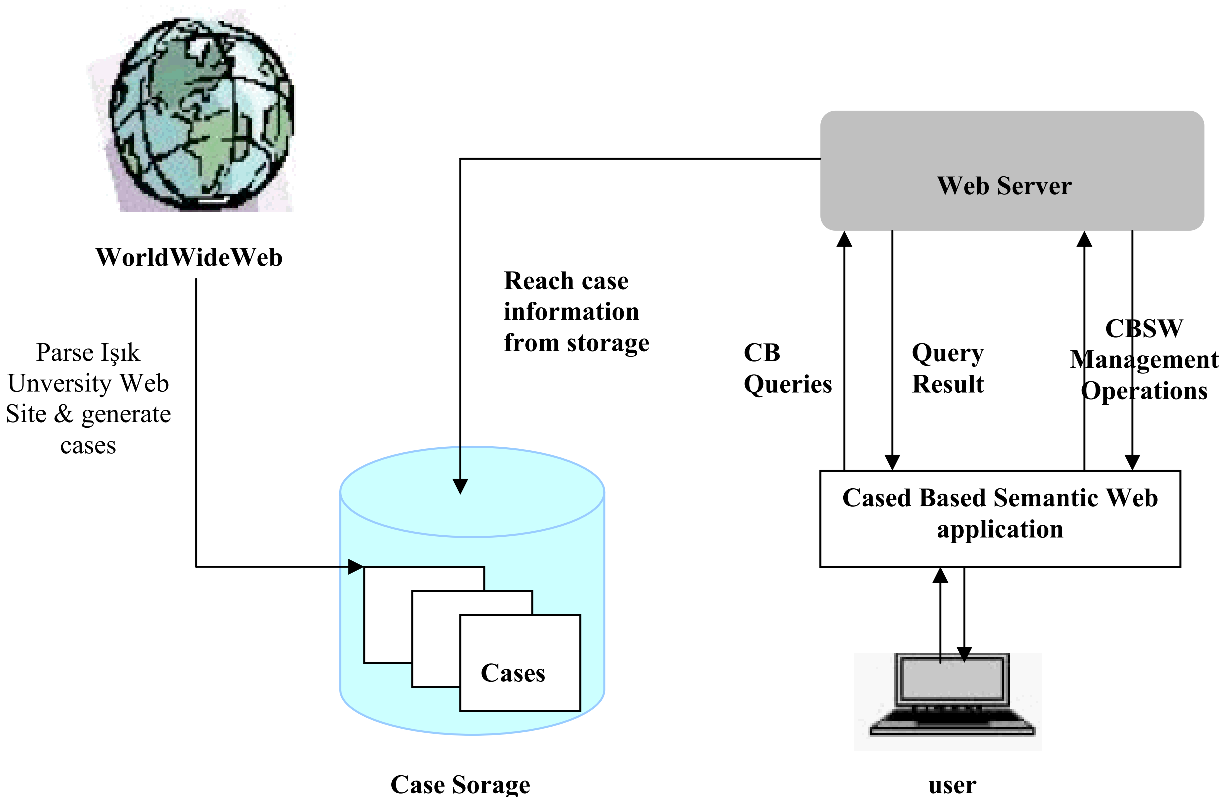

Figure 4.1 Architecture of proposed case based semantic web approach

Figure 4.1 represents the architecture of case based approach to semantic web. Cases are generated from the knowledge that stays at WWW and they are stored on a system like database or memory. Case based semantic web application is built with dynamic web pages on a webserver, which is integrated with case storage system. When user executes a query on system, case based application goes to case storage system, reads all knowledge and relation cases, and apply the relation rules on knowledge cases, and displays best results to user.

4.1 Representation Of Cases

Cases of CSWA application can be grouped in four.

- Knowledge Representation Cases
- Relationship Rule Definition Cases
- Interoperability Cases
- Experience Cases

Definition and example for four types of cases are described in this section.

4.1.1 Knowledge Representation Cases

These cases are defined for storing knowledge of web site in a case based approach. Figure 4.2 represents example of a knowledge representation cases.

Case # x

Name:	Synonyms
Value Pairs	
Computer Engineering	Selahattin Kuru
Computer Engineering	Ahmet Aksan
Electrical Engineering	Erdal Panayırcı
...	...

Figure 4.2 Example of a knowledge representation cases

4.1.2 Relationship Rule Definition Cases

Relationship between the information is defined in these cases. For example if a case is defined for case like A contains B, CSWA also knows that B is a part of A This logic is implemented at Case Based Semantic Web application. Figure 4.3 represents example of a relationship rule definition cases.

Case # x

Name:	Contains Relationship
Value Pairs	
Academic	Faculty
Academic	Institute
Institute	Art of Science
…	…

Figure 4.3 Example of a relationship rule definition cases

4.1.3 Interoperability Cases

Interoperability cases are used to link different domains in a world wide web. With interoperability cases it is possible to reach related information in different domains. Figure 4.4 represents example of a interoperability cases.

Case # x

Name:	Universities of Turkey
Value Pairs	
Işık University	www.isikun.edu.tr
Boğaziçi University	www.boun.edu.tr
Bilkent University	www.bilkent.edu.tr
…	…

Figure 4.4 Example of a interoperability cases

4.1.4 Experience Cases

Experience cases are used to store previous queries and their results to use them in for coming problems. System gains intelligency to point best matches with experience cases. Figure 4.5 represents example of a experience cases.

Case # x

Name:	Experience Cases
Value Pairs	
Keywords A	Case id x
Keywords B	Case id y
…	…

Figure 4.5 Example of experience cases

4.2 Technologies Used For Implementation

CSWA application composes of case storage system and web based case management system named Case Based Semantic Web application.

MYSQL is used for case storage system. The main reason that MySql is chosen as database is, it is free and fast RDBMS (Relational Database Management System) that can run on many operating systems - Windows, Linux, Mac OS, most varieties of UNIX (including Solaris, AIX, and DEC UNIX), FreeBSD, OS/2, Irix, and others.

Other than being priceless and fast system, MySql has many other strengths that make it chosen along it's competitors. It is secured with authorization system that allows some or all database privileges (for example, the privilege to create a database or delete data) to specific users or groups of users. MySQL can handle databases up to 50 million rows or more. So it can support large databases. For possible problems or troubles that you can live with system, technical support is widely available. A large base of users provides free support via mailing lists.

Cased Based SebWeb application is implemented as web based system to make it easily reachable. The popular web development language PHP is used for implementation of application. Apache Web Server platform is used to execute PHP program. Currently, Apache is the most popular web server software in the world. It offers a powerful web application environment. Integration of Mysql and PHP modules can be made easily. It is an open source tehnology, which means free and it’s available in various operating systems like Windows, Unix, Linux etc.

5. IMPLEMENTATION: A SIMPLE SEMANTIC WEB FOR IŞIK UNIVERSITY

Case Based Semantic Web application is implemented as web tool, to make it easily reachable. It is used to query and manage case based semantic web approach. The main page of Case Based Semantic Web application is displayed at figure 5.1.

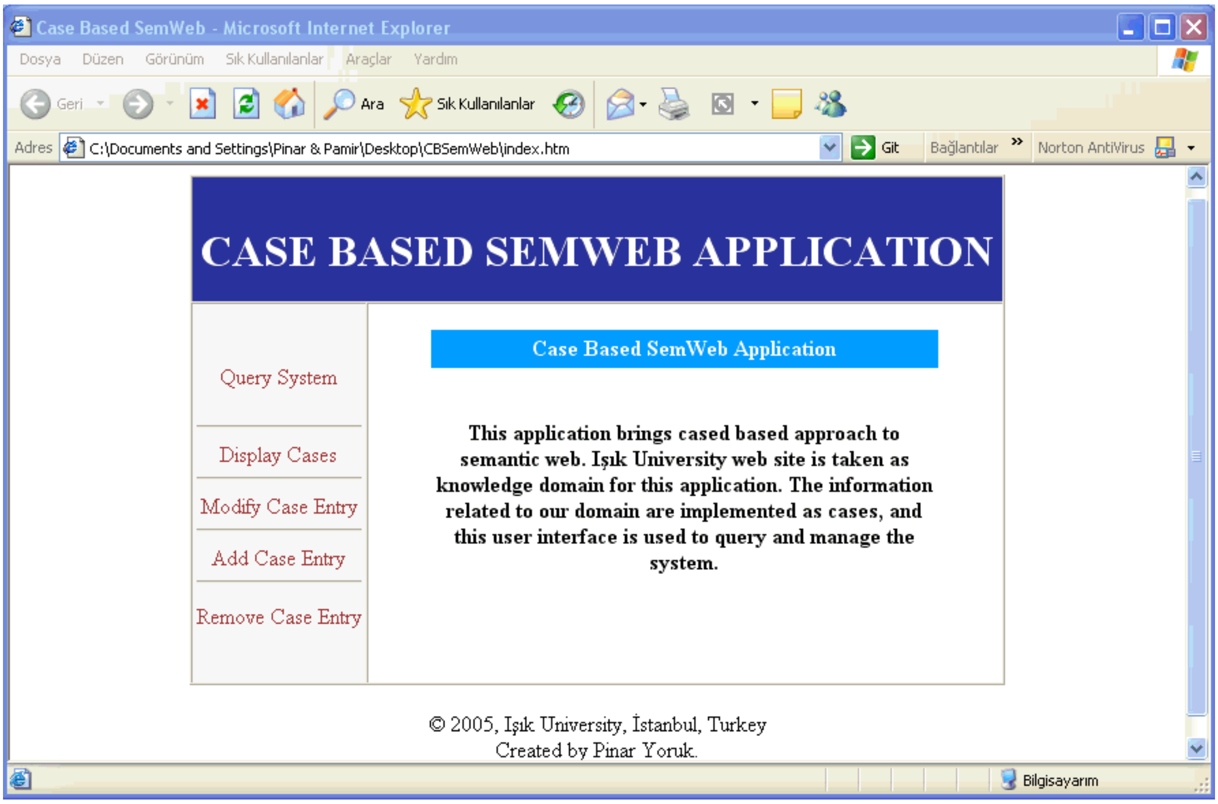

Figure 5.1 Main page of Case Based Semantic Web for Işık University

Case Based Semantic Web For Işık University contains query program to make search within the case system. Application is build on a dynamic structure and has management programs to extend, modify and delete cases.

5.1 Case Architecture of Case Based Semantic Web for Işık University Application

Case Based Semantic Web for Işık University application contains knowledge and relationship cases. Knowledge cases contain case information in it, while relationship cases hold the relations between the knowledge cases.

5.1.1 Knowledge Representation Cases

At this thesis we choose Işık University as our domain and implement this domain as a case base system.

Architecture of knowledge representation cases is displayed at figure 5.2.

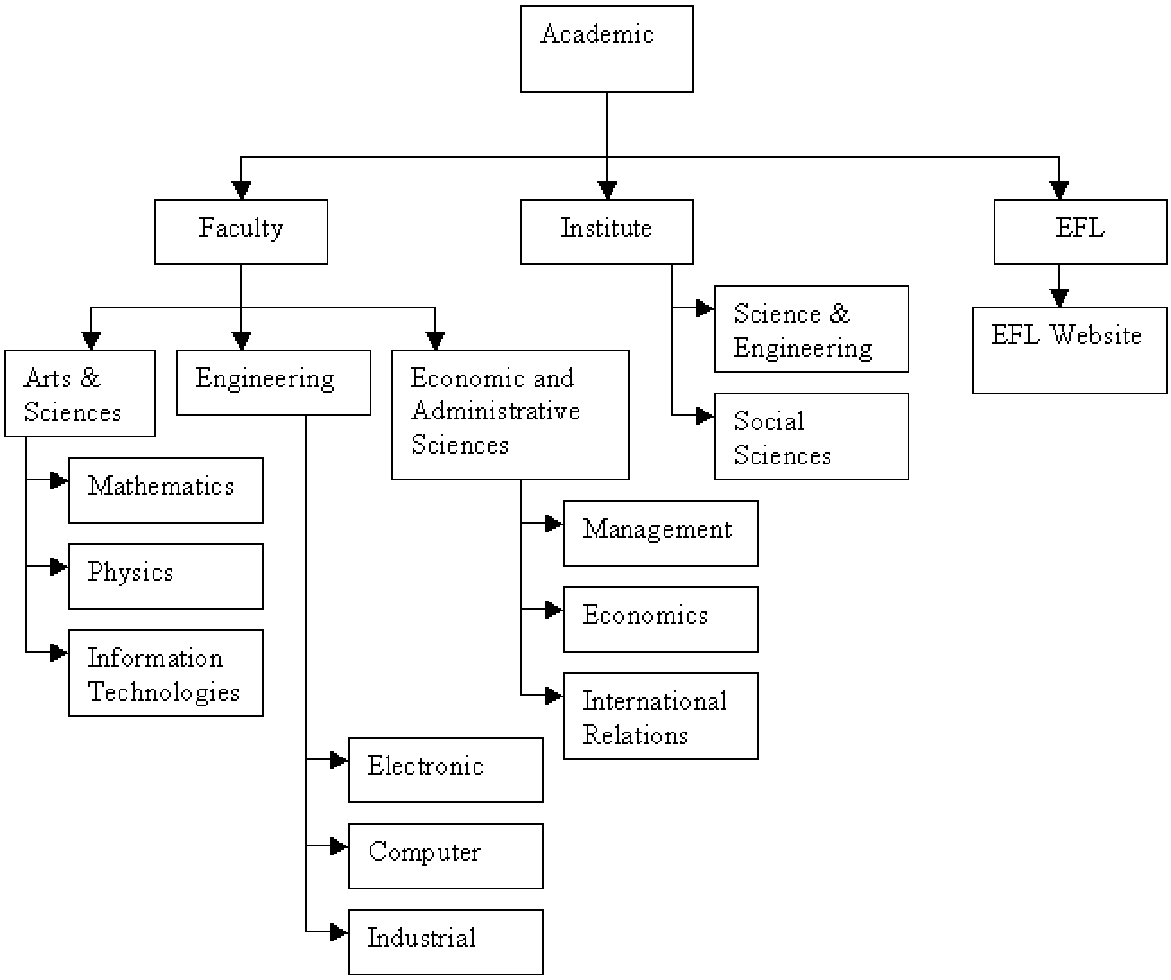

Figure 5.2 Architecture of Knowledge Representation Cases for Işık University

Every text box listed at figure 5.2 is implemented as case. Academic is taken as root case. Academic contains “Faculty”, “Institute”, and “English as a Foreign Language Departments” divisions.

For Işık University domain, we have three faculties: “Arts & Science”, “Engineering”, “Economic and Administrative Sciences”. Each of them is also defined as cases and they contain their department information. Institute case of Işık University, has two sub departments: “Science and Engineering”, ”International Relations”. Last child of academic is EFL and it stays as separate department.

5.1.2 Relationship Cases

Relationship between the knowledge representation cases, are defined at “Relationship Cases”. At Case Based Semantic Web for Işık University application, these two relationship cases are defined:

Contains Relationship:

Parent, child relations between the Işık University departments are stored at these cases. Academic is defined as root parent. Faculty, institute and EFL are children of academic. Faculty has three children named “Arts & Science”, “Engineering”, “Economic and Administrative Sciences”. This relation is defined at contains case as shown at figure 5.3.

caseid	parent	child
1	Academic	Faculty
2	Academic	Institute
3	Academic	EFL
4	Faculty	Engineering
5	Faculty	ArtScience
6	Faculty	Economics

Figure 5.3 Part of the contains case

When user search a keyword that stays as parent at “contains case”, system also shows information about its children. So although user didn’t enter exact keywords about its children, system would able to give extensive information about entered keyword. Same also valid for reverse relation; when user enters keyword that stays as child at "contains case", system also shows information about its parent.

Equals Relationship:

Equality case contains target words that can be misspelled at query time. For example instead of "department", user can enter "deparment". The systems that are based on keyword matches won't be able to understand this entry. At our system we define such relations at our equality relation and system become able to understand what user really wants to say.

Equality case also contains synonym information. For example user can search information with "teacher" keyword. The information that user searches, can contain "instructor" keyword, instead of "teacher" keyword. At our system we define such relations at "equality case" and by this way user would be able to find related information with each words.

World wide web contains information from different languages. User can't determine the knowledge that (s)he searches for, represented in which language. If (s)he works on systems that are based on keyword matches, (s)he has to make multiple queries with words from different languages. At equality case we define different versions of words for different languages. So when user enters a keyword in one language system would find corresponding word in other languages, and search will be done with those keywords also. And user would be able to find information that is related to entered keyword, although it is represented at different language.

Part of the equality case is represented at figure 5.4.

caseid	keyword	targetword
1	Faculties	Faculty
2	Fakulte	Faculty
3	Deparman	Deparment
4	Akademik	Academic
5	Enstitüler	Institute
6	Enstitu	Institute

Figure 5.4 Part of the equality case

Interoperability Relationship:

World Wide Web contains enormous size of data. It is impossible to put all this information in one group. Sub domains should be defined for organization of knowledge and relation between the domains should be obtained. At our system this relations is supplied with interoperability cases. When user enters a keyword that belongs to these cases, other domains that searched information can be found are displayed to user. So user can move from one domain to other, or one upper part. Part of the interoperability case is represented at figure 5.5.

←T→	caseid	keyword	domain	domainurl
	1	University	Bogazici	http://www.boun.edu.tr/
	2	University	Marmara	http://marmara.edu.tr
	3	University	Bilgi	http://bilgi.edu.tr

Figure 5.5 Part of the interoperability case

5.1.3 Experience Cases

Experience case is used keep current system updated. Once a matching case is retrieved, a case-based system should adapt the solution stored in the retrieved case to the needs of the current case. At our Case Based Semantic Web application for Işık University, we hold the best matching case entries at experience case, when and new best entry for a specific keyword is selected at the system; the best match is updated for that keyword. Part of the experience case is displayed at figure 5.6.

←T→	caseid	keyword	url
	3	faculty	http://www.isikun.edu.tr/english/index.php?table=h...
	4	Academic	http://www.isikun.edu.tr/english/index.php?table=h...
	5	engineering	http://muhendislik.isikun.edu.tr/muhf_eng/

Figure 5.6 Part of the experience case

5.2 Management System of Case Based Semantic Web Application

Case Based Semantic Web application for Işık University is a flexible system that user can add, modify and delete case information according to system needs. These modules give dynamic architecture to application. With stable case definition it won't be available to fit the system when information is changed.

5.2.1 Display Case

Display case option is used to show the cases and the information that is hold in these cases. When user clicks on "Display Case" link, the cases stored in system are displayed as shown at figure 5.7.

Figure 5.7 First page of “Display Case” option

First page shows the cases defined in system. If user want to see the information stored at specific case, (s)he can reach this information by clicking on specific case entry. The example for “Equality Case” is shown at figure 5.8.

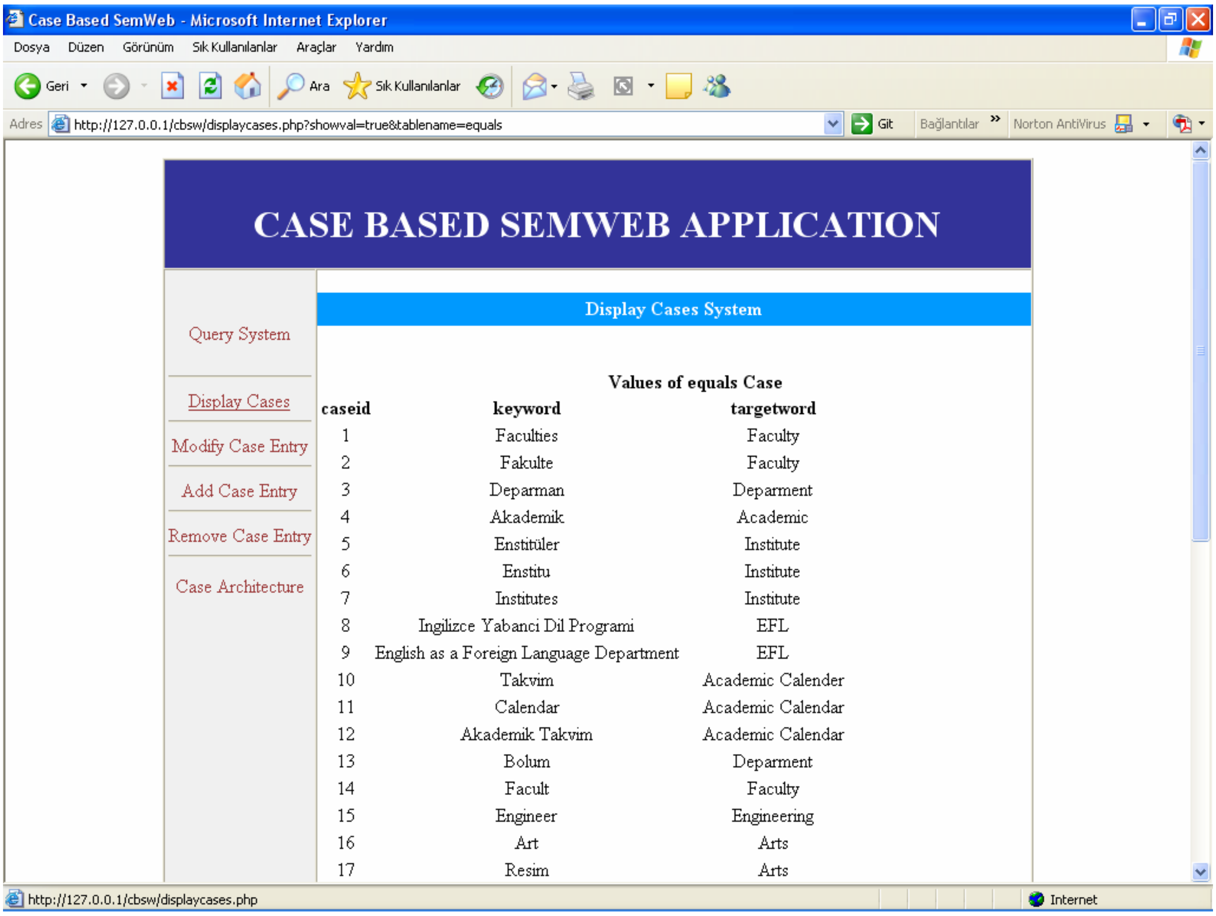

Figure 5.8 "Display Case" option page that shows the case information

5.2.2 Add Case Entry

Information stored in domain should be extensible. New information should be easily entered in system. Add case option is implemented for this aim. When user clicks the "Add Case Entry" link, cases defined at system are displayed. User selects the case that new entry will be added. Then user enters the data for new entry, and clicks "Add New Case Entry" button. Figure 5.9 shows the "Add Case Entry" option.

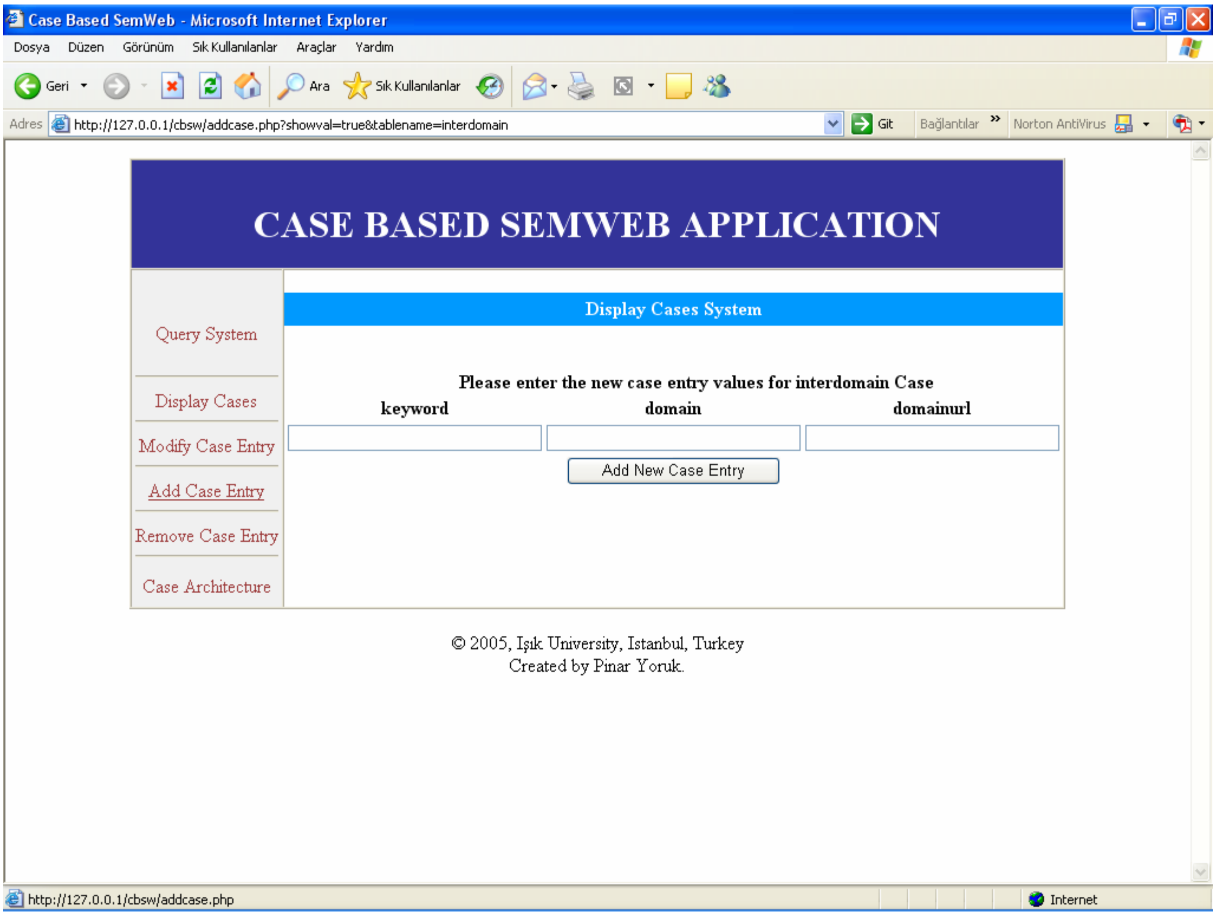

Figure 5.9 "Add Case Entry" option

5.2.3 Modify Case

Information in the knowledge domain should be easily updateable. Modify case option for Case Based Semantic Web for Işık University application is developed for this aim. At first page of "Modify Case" option, the cases defined at system are displayed. User selects the case that will be modified. Then the data stored in that case are displayed as shown at figure 5.10.

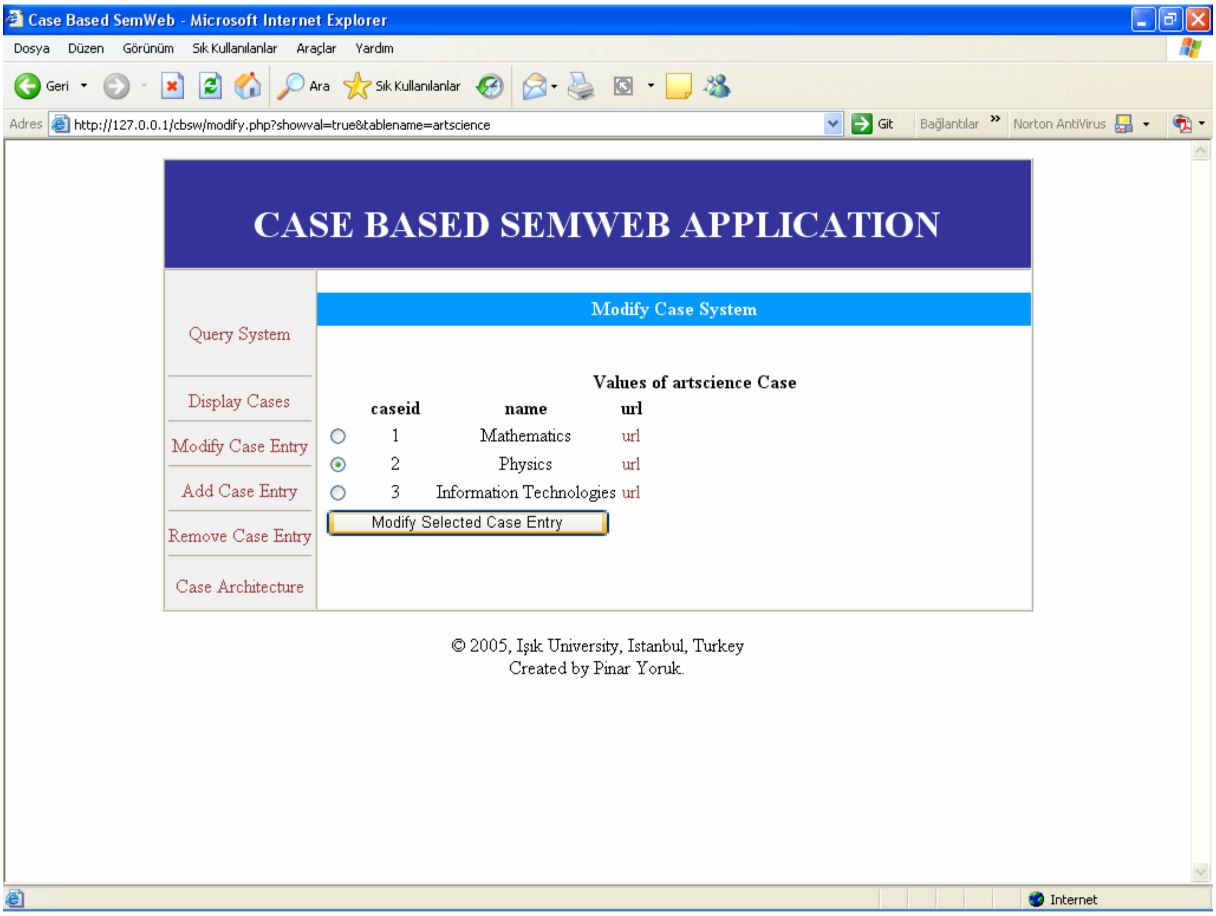

Figure 5.10 First page of "Modify Case Entry" option

User selects one of data to modify and clicks "Modify Selected Case Entry" button. After clicking the button data will be displayed in an updateable form. User changed the data as needed and clicks "Modify Selected Case Entry" button to completed case modification. Last page of "Modify Case" is shown at figure 5.11.

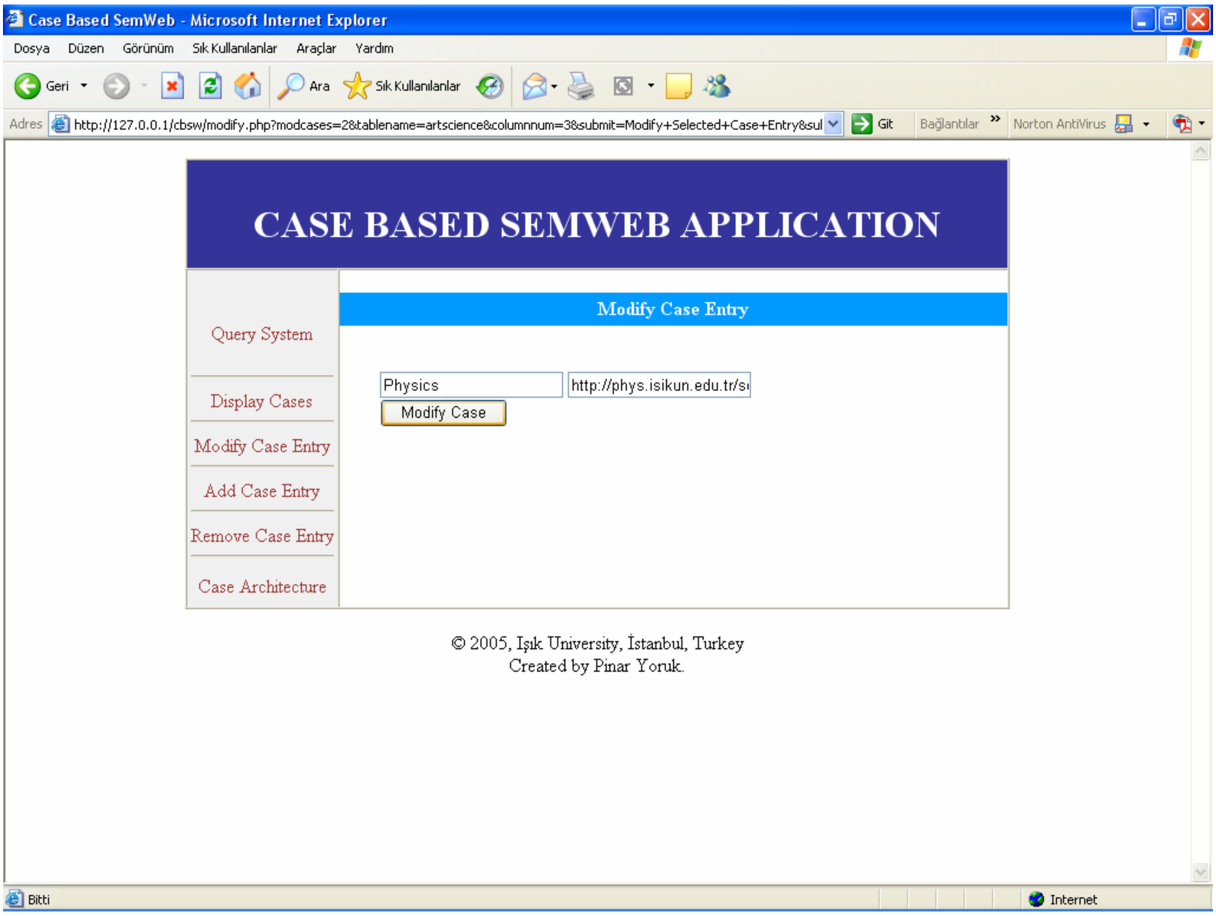

Figure 5.11 Second page of "Modify Case Entry" option

5.2.4 Remove Case Entry

Işık University is a dynamic domain that knowledge can be changed very frequently. For example new lessons can be added, or removed. "Remove Case entry" is developed to easily delete case data from system.

At first page of "Remove Case Entry" option cases stored in the system are displayed. When user selects one of the cases, the entries stored at that case are displayed with a checkbox to delete needed ones. As shown at figure 5.12, selecting the case entries and clicking the "Remove Selected Case Entries" button, complete the operation.

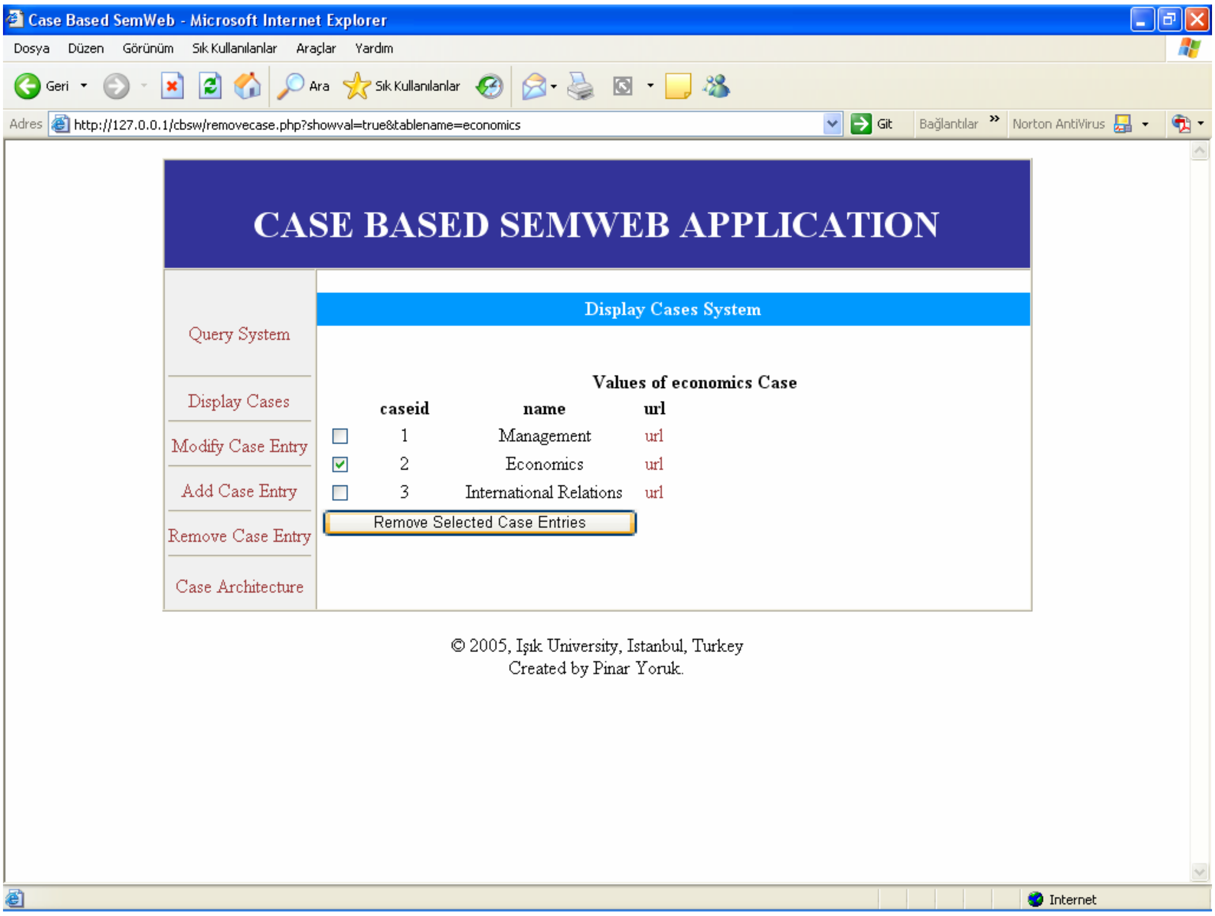

Figure 5.12 "Remove Case Entry" option

5.3 Query System Of Case Based Semantic Web For Işık University

Query part of the Case Based Semantic Web application for Işık University, is used to define and execute queries among case objects, stored at database. The difference of case based query system from current query systems is, it is not only based on the keyword matching. It uses the relation cases to gain more intelligence and correctly understand user needs. Snapshot of query system is shown at figure 5.13.

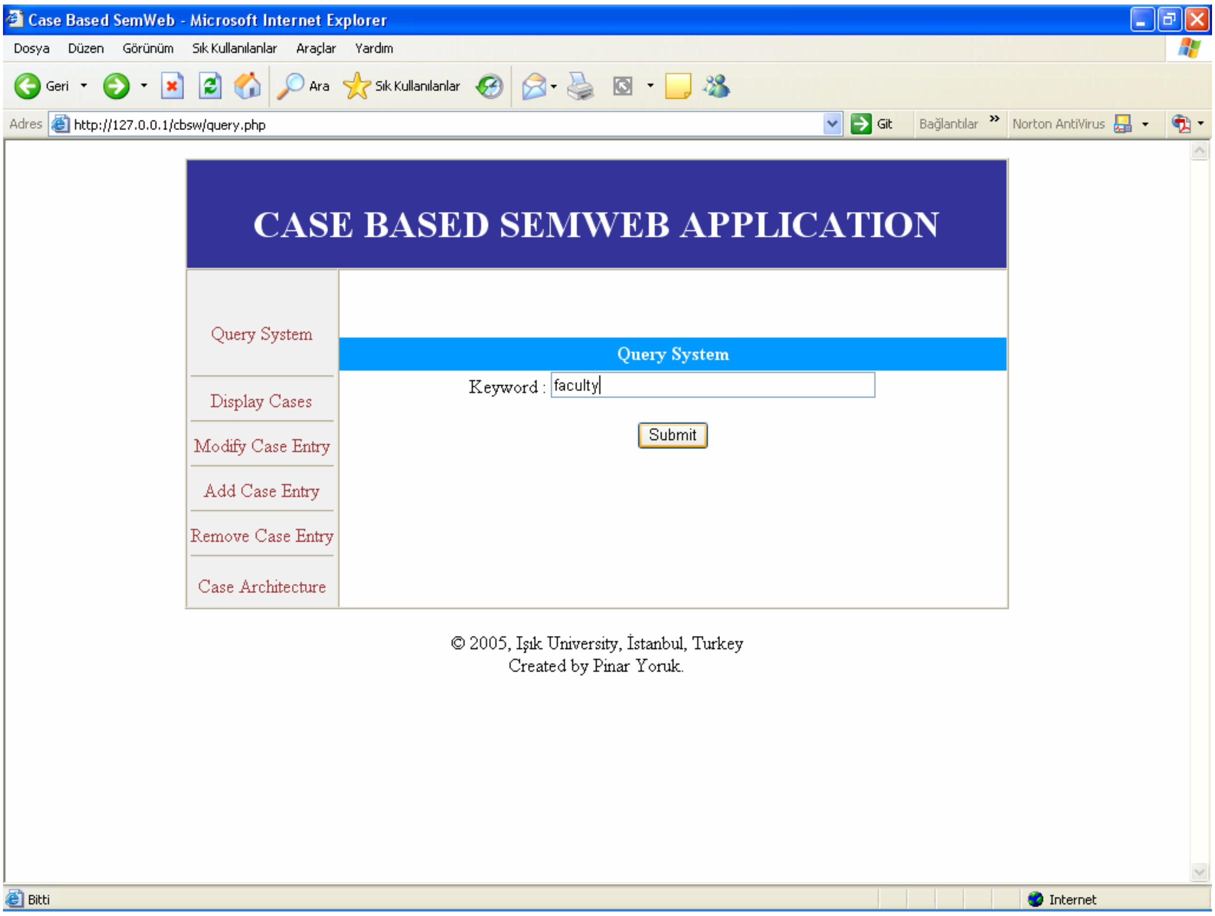

Figure 5.13 "Query Case Entry" option

Working mechanism of case based semantic web application is like this:

- User enters the keywords that will be searched. At first query system reads the equality cases to find out other keywords that can be used at the search. Equality case contains synonyms, corresponding version of entered keyword at other languages, and targeted word if the keyword is misspelled. If new entries are found that matching the entered keyword at equality case, then they are also added as search keywords.
- Experience case is searched to find the matching entries with the entered keywords.

- All the knowledge cases stored at the system are searched with these keywords. The matching entries are scored, according to how many hits is made to keywords and sorted in decreasing order.

- There can be cases that doesn't contain case name at stored entries. To avoid not showing values of such cases, we implement showing the data of the case, if the entered keyword represents a case. For example user can enter "engineering" as keyword. At system there can be a case named engineering with data containing: "Computer", "Industrial", "Electronics" and their related URL without engineering keyword. If we only look at the data stored in the case then we won't show entries of this case. We avoid it but checking as if the keyword matches the case, and if there is a match we show the data of the case.

- Query system also checks the "contains" relation case, and if entered keyword contains parent or child, these entries are also displayed to user.
- Lastly query system checks the interoperability cases as if entered keywords can be related to other domains. If there is a match, system shows the URL of the domains, and user can move to that domain by clicking on the link.
- When the query is finished at the background results are displayed to user.

Results of the query made with "Fakulte" keyword is displayed at figure 5.14 and figure 5.15. At this case keyword is entered in Turkish. Information stored in our domain is in English. Firstly system checks the equality case to find matching keyword at system language. It is found as "Faculty" and search is made on in.

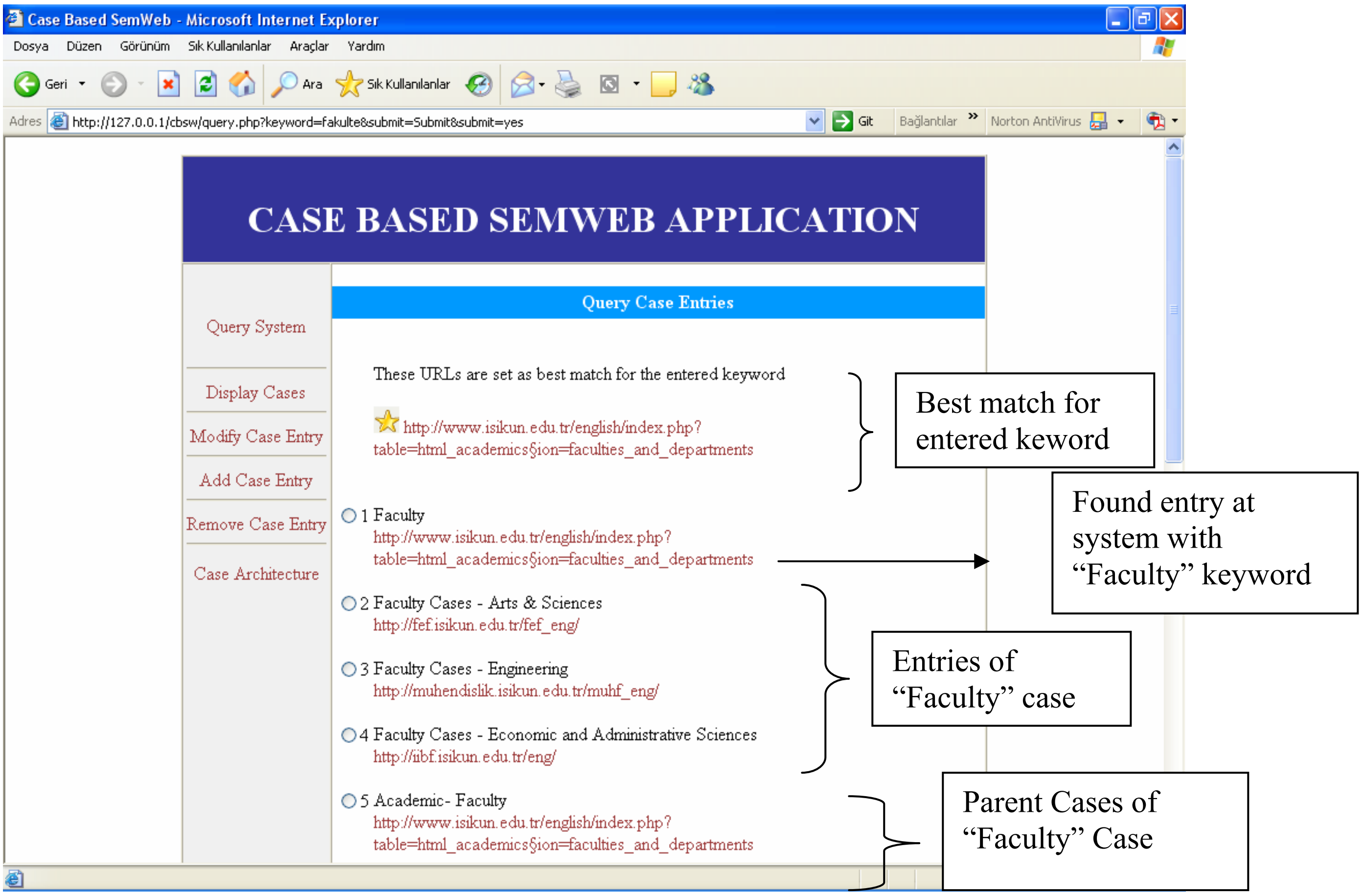

Figure 5.14 Result of case based query (first page)

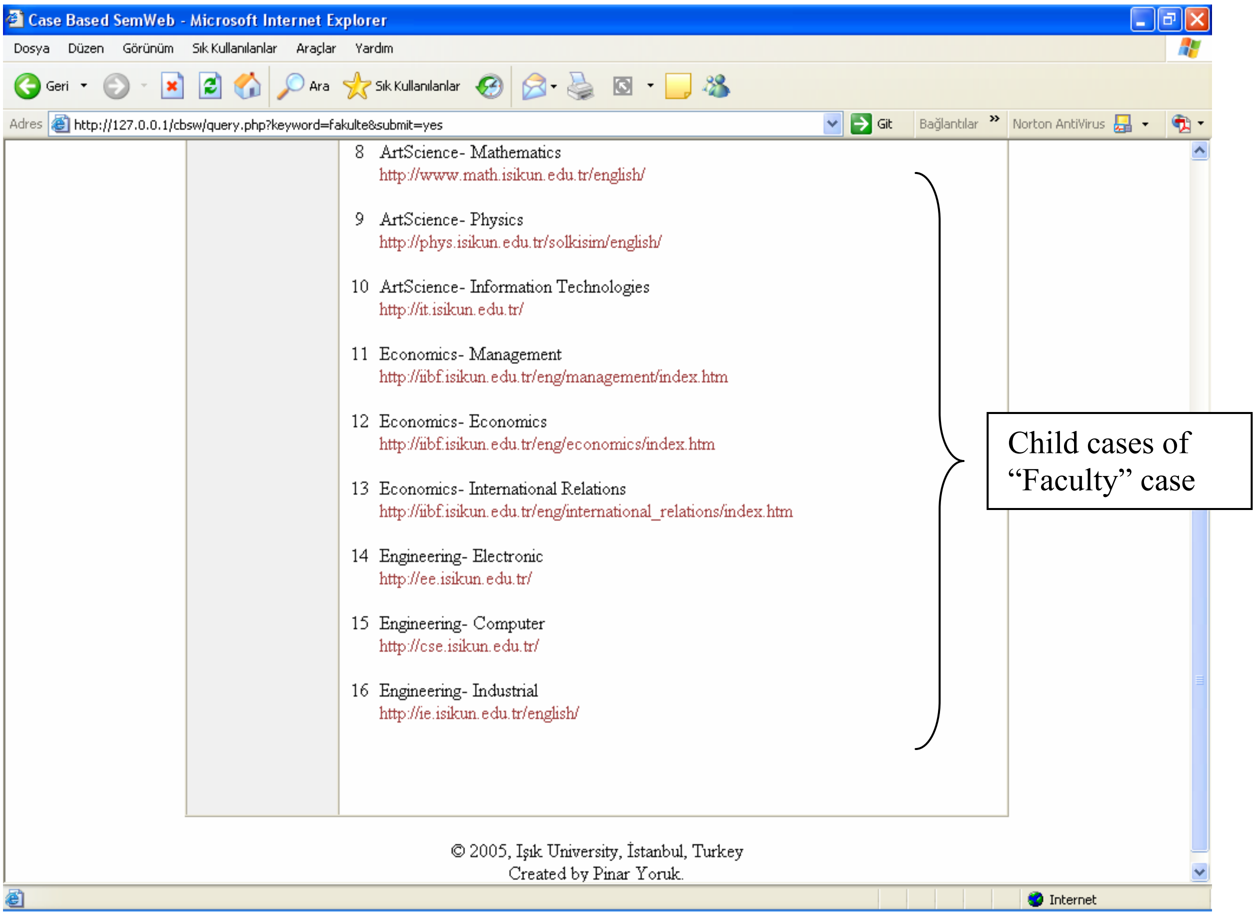

Figure 5.15 Result of case based query (second page)

6. CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

World Wide Web contains enormous size of data. New approaches are required to use these information effectively. In this thesis we propose a case based approach to semantic web. We explain how semantic implementation is done with this new approach. We also realize the approach by implementing a Case Base Semantic Web application.

Case based approach brings new look to semantic web. To date, ontology based systems are used for semantic web. But differences at ontology standards and complexity of relating document with ontology annotation, cause difficulties at implementation of this approach.

At case based approach of semantic web, knowledge are implemented as cases. Special case implementations are done for rule definitions. With this approach searching is not done, only by finding matched strings. Rules are processed and applied on knowledge cases to find best matching criteria. By this way, system gained intelligency. Other special case implementation is used for interoperability functionality. With these case implementations, system gains ability to interact with other domains.

As a future work case library can be implemented for a semantic world. Generally many different cases should be implemented, to cover large portion of World Wide Web. Another direction for future growth will be extending case based approach to separate domains for other purposes (e.g. e-commerce).

To ensure distributed architecture for case base semantic web applications, investigations can be done for exchanging knowledge between different domains. Things to be considered in this area are efficient data retrieval, easy maintenance and providing efficient transmission over a network. Case base markup language can be defined and used for this aim. This representation language could be used by all case based semantic web applications that exchange information across an Internet or Intranet. This language can be based over XML, as a metalanguage for representing structured data over network systems.

Performance and search power comparison of different approaches to semantic web can be also handled as future work.

REFERENCES

[1] Alterman, R. (1989). Panel discussion on case representation.In, Proceedings of the Second Workshop on Case-Based Reasoning, Pensacola Beach. FL, US.

[2] Birnbaum, L. & Collings, G. (1989). Remindings and Engineering Design Themes: A Case Study in Indexing Vocabulary. In, Proceedings of the Second Workshop on Base-Based Reasoning, Pensacola Beach, FL.

[3] Schank, R. (1982). Dynamic memory: a theory of reminding and learning in computers and people. Cambridge University Press, Cambridge, UK.

[4] Falkeneheimer, B., Forbus, K.D. & Gentner, D. (1986). The structure mapping engine. In, Proceeding of the Sixth National Conference on Artificial Intelligence, Philadelphia, PA, US.

[5] Kolodner, J. L. (1993). Case-Based Reasoning. Morgan Kaufmann.

[6] Merseth, K. K.. "The Case for Cases in Teacher Education". Paper presented at the annual meeting for the American Association for Higher Education (AAHE) and the American Association for Colleges of Teacher Education (AACTE): Washington, D.C., January 1991

[7] Cooper, J. M., and R. F. McNergney. "Introduction: The Value of Cases in Teacher Education." In J. M. Cooper (Ed.), Teachers' Problem Solving: A Casebook of Award-Winning Teaching Cases. Boston: Allyn & Bacon, 1-10. 1995

[8] Kleinfeld, J. (1991). "Changes in Problem Solving Abilities of Students Taught Through Case Methods". Paper presented at the annual meeting of the American Educational Research Association (AERA): Chicago.

[9] Ip, A., & Naidu, S.(2001) Experienced-Based Pedagogical Designs for eLearning in Education Technology vol XLI No 5, September-October 2001 pp53-58 Magazine for Managers of Change in Education, Publisher: Educational Technology Publications

[10] "OWL Web Ontology Language Overview", W3C Recommendation 10 February 2004, http://www.w3.org/TR/2004/REC-owl-features-20040210/.

[11] W3C, "Extensible Markup Language (XML) 1.0", Third Edition, http://www.w3.org/TR/REC-xml/ , 2004.

[12] "RDF Primer", W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-primer/ .

[13] Michel Klein, "XML, RDF and relatives", IEEE Intelligent Systems, 2001.

[14] World Wide Web Consortium, Namespaces in XML, January 1999. http://www.w3.org/TR/1999/REC-xml-names-19990114/

[15] Ora Lassila, and S. R. R. (eds), Resource Description Framework (rdf) model and syntax specification. W3C Recommendation, February 1999.
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[16] Paulson, P.; Juell, P.;Potentials, IEEE ,Volume: 23 , Issue: 1 , Feb.-March 2004 Pages:31 – 33

APPENDIX

CD containing thesis text and code of the program is attached.