Intelligent Integration of Information From Semi-Structured Web Data Sources on the Basis of Ontology and Meta-Models

Guntis Arnicans
University of Latvia
Riga, Latvia

Girts Karnitis*
University of Latvia
Riga, Latvia

Abstract — As computer users face an increasing amount of various semi-structured information sources, the issue of correlating, integration and presenting related information to users becomes all the more important. As a solution, we propose the Semi-Structured Data Universal Data Browser, which, in its operations, makes use of descriptions of data sources that are presented in the form of meta-models or ontologies, ensuring the user’s ability to use information from the data sources. Information from semi-structured data sources is analyzed, transformed and stored in the Semi-Structured Data Universal Data Browser database, which is based on meta-models. The ontologies of information in each data source are preserved, and they are mutually linked to logical and global ontologies through the use of mapping. As an example, we use the integration of information on Internet homepages about products and their classifications.

I. INTRODUCTION

In the modern world, there are countless information systems of many different kinds. Most are integrated with other systems. Users increasingly need to integrate information that comes from a variety of information systems. Integrating major systems in which functions and stored information change seldom — that is far simpler than integrating systems, which are very heterogeneous, which change over the course of time, and which exist independently of other systems. In the first group there are legacy information systems, which are based on relation databases. In the latter we find the Internet homepages of many companies — sites where companies provide information about the products and services that they offer.

The authors of this paper have had to deal with the issue of integrating a number of registers of national importance. An Information System Universal Data Browser (IS UDB) was set up, and it provided for the merger of information from all of the registers. The information could then be reviewed in a consolidated way — as if just a single system were being used. The work was made much easier by virtue of the fact that information from the registers was accessible through special services and was well structured [1][2][3][4].

Many systems, including the IS UDB, are intended for work with structured data. At the same time, however, information on Websites and elsewhere is stored in semi-structured form. “Semi-structured data is all the digital information that cannot be easily and efficiently modeled using traditional schema tools, software, or methodologies.” [5] There are many different and even unique methods for processing semi-structured data. For instance, a specific type of processing is needed for every Web page.

Creating and using the IS UDB proved to be relatively simple, and that created a question: Could the same browser be used to consolidate and review semi-structured information?

In this paper authors propose a possible solution for the extraction, processing, classification and display of semi-structured information. The solution is based on the authors’ experience in setting up an IS UDB.

The authors selected semi-structured sources of information for this experiment — company homepages that describe goods and services. The potential user of the browser is a consumer who wishes to receive quick and consolidated information about the desired product, or who wishes to search for a product of the desired type.

We propose that a modification of the IS UDB can be used in this case. Whereas information systems, which usually offer services for information search and retrieval, the situation with Web pages is that the services are not offered. All that is possible at first is a full selection of Web pages from the data source. Then there can be extraction, analysis and storage of the data in the IS UDB database in structured format until such time as the information source offers better opportunities.

* This work was funded by grant from the European Social Foundation (W-ESS3-007).
II. AN ELECTRONIC CATALOGUE OF PRODUCTS

In order to look at the issue of how an IS UDB can be adapted to the integration of partly structured information, let us select a fairly classic example – integration of electronic catalogues of products. Sources of information in this case can be Internet shops, as well as the homepages of manufacturers.

If we wish to find the product that we desire and purchase it at the best possible price, then we probably will not be happy to find that we must browse through thousands of Web pages to see whether the necessary information is there. It goes without saying that comparative information in this case is not assembled in an easily surveyed table. We must consolidate the information ourselves. In such cases, a domain-specific tool is needed, which allow people to receive information about products from all data sources, or at least many data sources (Internet sources, as well as the homepages of manufacturers, retailers and service providers).

Throughout the world, Internet homepages on which manufacturers, retailers and service providers publish information about their goods and services (i.e., their products) are becoming more and more popular. More than 20 online stores sell household electronics in Latvia, and this number is on the rise. It is hard to obtain information from different information sources, however, because each appropriate page must be found, opened and analyzed. The user could receive information about the desired product in various ways.

First of all, one can visit the homepage of the manufacturer. It usually offers all information about the company’s products, but very seldom are there comparisons with the products of other manufacturers. It is not possible to compare prices offered by various retailers of the product.

Second, one can visit specific stores, but that takes a lot of time to compare offers from number of stores.

Third, there are special homepages which collect the offers of various retailers and offer price comparisons – eBay, NexTag etc. In different regions of the world, however, there are different levels of IT use and different economic justifications for the establishment of centralized services. There have been some attempts in Latvia to offer such services. Usually such pages in Latvia offer primitive information about a narrow range of products and their features.

Fourth, information can be sought out via a text-oriented search service such as Google and Yahoo! Search engines are also of no use, because when one seeks out a specific product, the result is not only information about the availability of the product through Internet shops, but also information about other pages on which the product is mentioned. The number of selected pages is enormous.

Fifth, it is possible to use domain-specific search engines which use various methods of classification and limit the number of homepages that are found.

A review of the various commercial services which offer price comparison shows that information about products and prices is delivered to these services by the retailers themselves, making use of electronic methods for that purpose. This means that there is a thankful subject for research. We can check what can be obtained from homepages without the delivery of structured data, looking to see whether the same level of functionality and quality can be obtained as is the case through commercial services. Perhaps in some areas, the functionality and quality are even better.

Analysis of a series of Internet shops in Latvia and abroad makes it clear that the structure of Internet shops tends to be quite similar. Internet stores have a fairly vast and multi-level hierarchical classification, and each class, even the most specific one, has a series of products. Classification varies from one Internet shop to the next. None of the classifications of Internet shops can be said to be correct or incorrect, because each was developed on the basis of sufficiently logical considerations. There tend to be problems of heterogeneity among the classifications:

- The same class has different names in different places;
- The same class is divided up into subclasses in different ways (sometimes it is not divided up at all);
- There are several classes with the same name.

Each product has a series of attributes, which describe it. Products in a single group in a single store tend to have the same attributed, while products in different groups have different sets of attributes. A single product can have different attributes in different Internet shops:

- One shop presents the object’s attribute, the other does not;
- The same attribute is identified differently;
- The same attribute contains different values;
- The same attribute contains different kinds of data.

III. INTEGRATION OF INFORMATION

What do we mean in our work when we use the concept of “integrating information,” and how do we wish to ensure that integration? Our goal is to use established technologies to create a tool which allows ends users to receive unified information from various heterogeneous sources of data, ensuring the necessary data transformations and making use of concepts that are understandable to the user. The user of the tool can choose his or her own unified view of the data that are available.

Data integration can be reviewed at various levels of complexity. In our understanding, the first is the physical level, where all data come from a single data source. We do not transform the data; we just arrange them in various combinations or views. These functions are handled by the Simple Database Browser that is reviewed in our work.

The logical level is the second one. Data still come from a single source, but they are received with the help of wrappers. The mechanism of views or mediated schema is used. The Logical Database Browser handles these functions for us.
The next level is the system integration level, where data are extracted from a multiplicity of data sources. These functions are handled for us by the Information System Universal Database Browser. It is based on the principle that information from each data source is received on the basis of the Logical Database Browser principle. Integration is achieved with the help of a simple ontology, one which links several of the most important entities and attributes among various sources and requires the translation of the values into a unified format.

The aim of the paper is to offer a solution for the integration of semi-structured information. The proposed technology is based on the widespread use of local ontologies, describing each data source as a part of this. Global ontologies are also established – ones which link local ontologies and offer various views of one and the same data. The Data Warehouse principle is also used – semi-structured data are translated and preserved in the tool’s database as structured information. This enables integration of information and a survey of information on the basis of the already tested Information System Universal Database Browser.

IV. USE OF ONTOLOGIES IN INTEGRATION OF INFORMATION

Ontologies were established and developed so as to make it easier to deal with correlated knowledge, to exchange knowledge, and to put it to effective use in the context for which it was needed. Ontologies began to develop most rapidly when they became more popular in dealing with issues such as natural language processing, depiction of knowledge, integration of intelligent information, integration of information systems, extraction of information from various data sources, and improvements to the world of E-commerce. “An ontology is a formal, explicit specification of a shared conceptualization” [6]. Ontologies became popular because they offer a common understanding of a concrete domain, and they ensure good communications among people and systems.

In wikipedia an ontology for computer science is defined as “In computer science, an ontology is a data model that represents a domain and is used to reason about the objects in that domain and the relations between them.” Ontologies generally consist of: 1) Individuals (the basic or "ground level" objects), 2) Classes (sets, collections, or types of objects), 3) Attributes (properties, features, characteristics, or parameters that objects can have and share), 4) Relations (ways that objects can be related to one another).

If we look at the homepages of various companies, we can see similarities in terms of the way in which information is offered about the products that are on offer. Basically there is classification of products and information about the products themselves.

In order to link multiple data sources and the information contained therein, we use many ontologies (Fig. 1).

Fig. 1 depicts a set of many linked ontologies. It would be simpler to work with one or two ontologies, thus making the processing model simpler. However we choose to make a set of ontologies where each of ontology has different role and contains different kind of information.

We can say that each user has his or her own ontology when it comes to a sphere that is of interest to the individual. That is illustrated if only by the fact that nearly every page has its own classification, i.e., it depicts the “ontology of the designers of the page”. It means that we must allow a number of ontologies of each type.

The ontologies of data sources must not be ignored. That means that when a user is offered search and browsing opportunities, we must try to use the structure and terminology of the ontologies.

Let us introduce three levels of ontologies on the basis of their meaning and their purpose – the physical, logical and global level.

The physical level describes the structure and syntax of information that has been placed into the source. There is a notation of those components which contain the data that we need. The structure whereby data are preserved (individual information units, lists, tables, pictures, hyperlinks and links among these various elements are linked) is discovered. There is no attempt to understand the semantics of the information. Data that are received are very general – string, URL, picture.

The logical level describes the structure and syntax of information that has been placed into the source. There is a notation of those components which contain the data that we need. The structure whereby data are preserved (individual information units, lists, tables, pictures, hyperlinks and links among these various elements are linked) is discovered. There is no attempt to understand the semantics of the information. Data that are received are very general – string, URL, picture.

The logical level describes the semantics of the data source. The necessary information is extracted from the texts. It is transformed to the desired data type and then linked together in a logical structure. The job at this level is to depict the information form the data source at the highest level of generalization while preserving the logic that was used in the data source. At the same time, it is necessary draw closer to the description of data objects that has been
defined at the global level and to the relationship among those standards.

The global level describes the standards for products and their classification standards or patterns. Here the aim is to link together information that is stored in the various data sources.

In addition to the ontologies that are described here, we can dynamically establish virtual ontologies, because the global level ensures the linkage of ontologies that exist at far lower levels on the basis of various aspects.

For instance, a potential method, which should be tested in practice, is the merger of many physical ontologies into a single ontology. Similar ontology nodes are brought together, and all names are left in place so that they can be identified. This basically produces a graph, and users must be given a way to maneuver through it easily so as to find the appropriate node.

Further in this chapter we will take a look at the essence of each ontology and why it is necessary.

A. Classification of products

Most companies use classifiers to classify their products. A review of homepages shows that there are many different principles for classification, and each company has its own ideas about the “right way” to do this. Only when it proves difficult to co-operate with other partners do companies seek out unified standards.

If we want to integrate information about products, then we have to think about the classification and reclassification of each data source on the basis of one or more global classifiers or standards [7][8]. In E-commerce, there are popular standards such as UNSPSC and ECLASS, or we can turn to the classification system used by a major and popular company such as eBay. We can also establish our own classification if we do not find a ready-made one that is appropriate for our purposes.

For instance, we wish to extract information about the Nokia 6230i and Nokia 7260 mobile phones, which have been recommended to us by friends. We are also interested in how convenient it is to use these mobile phones while we are in a car. Fig. 2 shows the various classifications that have been obtained from the manufacturer (Nokia), a large online marketplace (eBay), a small E-store (PriceList), and a product classification standard (UNSPSC).

The classifications “Nokia”, “eBay” and “PriceList” serve as a foundation for establishing separate physical and logical ontologies for each data source. For each data source, we analyze and record a classification with the structure and content that appears on the relevant homepage (Physical Classification), and then we establish mapping between this classification and the local logical ontology (Logical Classification).

There can be several global ontologies simultaneously, because the global ontologies of various areas of business can differ. For instance, manufacturers of products may use a different ontology than the retailers of the same products. In order to move from one sphere to another, we need mapping among the global ontologies.

Global Classification types of ontologies are used to integrate the ontologies of different data sources. What is necessary here is to establish mapping between each Logic Classification ontology and each Global Classification ontology.

For instance, one of the global ontologies can be the UNSPSC standard. When we review the Nokia 610 Car Kit, we can see that it is in line with several classification positions from the UNSPSC standard. In a different classification, the situation might be different.

In [7], there is a look at the most important things that are needed in order to ensure mapping among classifiers and various ontologies. The most important relations among ontologies are based on the following types of mapping:

- **SYN** (synonym of) – two product classes are equivalent;
- **NT** (narrower classes) – one class is a sub-class of the other;
- **RT** (related classes) – there is a relationship among classes which links them together in one context or another.

B. Ontology of products

Product description is another typical element on homepages, which essentially can be described through its own ontology. Each product usually has a description of its properties and various aspects of its use. For instance, a phone is described by listing its Band (EGSM 900, GSM 1800, GSM 1900), Size (weight 99g, 103 mm * 44 mm * 20 mm), Display (Active TFT color, 65536 colors, 208 * 208 pixels), Battery (BL-5C, talk up to 3-5h, standby tipe up to 150-300h), pictures, etc. Related information for the given
product may include compatibility with other devices and interfaces.

Each company establishes product ontologies on the basis of its own considerations. Ontologies differ in terms of structure – a flat structure (a list with pairs {Attribute, Value}), or a branched structure (the pairs {Attribute, Value} can be grouped in logical groups) [8]. It is easier to integrate if there is not a branched structure, but in that case the user loses the opportunity to operate with mutually linked information.

Thus, for instance, Nokia describes many of its phones on the basis of the structure that is shown in Fig. 3. This structure can serve as a Physical Product ontology for most or all of the phones that are shown on the Nokia homepage. In describing the structure, it is desirable to use concepts that are close to those which are used in HTML (list, table, hyperlink).

The semantics of the description are noted with the help of the Logical Product ontology. Fig. 4 shows a fragment from the ontology of the Nokia public phone. In terms of physical ontology, the list of symbols “Size”, “Weight: 99 g (including battery BL-5C)”, “Dimensions: 103 mm x 44 mm x 20 mm, 76 cc”, “Display”, “Active TFT color display is noticeably brighter”, “Supports up to 65,536 colors within 208 x 208 pixels” corresponds to the Nokia 6230i. The value “99”, “103”, “44”, “20”, “76”, “Active TFT color”, “65,536”, “208 x 208”, for its part, corresponds to the logical ontology.

Differences among product ontologies and opportunities to integrate them are described in [9][10]. Sadly, integration and standardization are quite problematic, because companies in a single business area usually are not sufficiently interested in standardization – after all, they are competitors.

If we are to avoid a situation in which mapping must always be defined manually, we can analyze the structure and text so as to engage in semi-automatic definition of mapping. We can use the following methods that have been developed for the integration of ontologies [11]: Text similarity, Keyword extraction, Language-based methods, Identification of word relations, Type similarity/Domains and ranges, Class inheritance analysis (is-a), Structural analysis/Taxonomic structure, Data interpretation and analysis of key properties, and Graph-Mapping.

As is the case with classification, products also have several ontologies – Physical Product describes a single type of products on the basis of the logic of page designers, Global Product is a global product ontology, which links various data sources, and Logical Product ensures transfer among the aforementioned ontologies.

In order to ensure that integration is as complete as possible, mapping is established between Global Classification type ontologies and the Global Product ontologies. In order to make it easier to search for products on the basis of their properties, the Global Attribute ontology is introduced. It helps to link or classify the attributes of one kind of product – Price, Weight, Manufacturer, etc. For instance, the global attribute “Company” that is depicted in Figure 8 links all of the products which have the relevant attributes – this belongs to the attribute class “Company” (in our case, it is the attribute “Manufacturer” in both product classes).

Relations among objects are described with the help of Global Relation. In our example, this links mobile phones with appropriate accessories and vice versa (Fig. 5).
A Universal Data Browser (UDB) is a tool that allows one to search and survey data and integrated information on the basis of the meta-models, ontology and ways of depicting information of data sources. The architecture and operations of a UDB can vary on the basis of the purpose for which it is used.

These are the main operating principles of a UDB [3]:

- The data source is described via a formal data model;
- Data can be accessed via special functions (wrappers);
- The depiction of information is based on special patterns or templates which are defined for various sub-sets in the data model and which then present the data in the semantics which are required by the user;
- Various types of ontology are used to link various data objects and to integrate information from one or more data sources.

Let us now survey various ways of using the UDB and the way in which we have evolved from a browser in just one database to the integration of many, heterogeneous data sources and a browser of consolidated information.

### B. The Simple Database Browser

The Simple Database Browser is the simplest type of UDB [12]. It allows users to survey data from a specific database. The main aim is to offer equipment to the user which ensures the establishment of a database survey system from the physical data model that is described with the ER diagram. Thanks exclusively to our model, the developer obtains a rapidly generated database survey system, a simple information analysis and filtering tool, a data entry and editing tool, a prototype for a far more serious business application, as well as a simple database testing tool.

There are several defined screen templates which allow us to manage data in a database. By the term screen here we refer to the application window, the survey page or even the frame of the survey page. The main principle of screen generation is to establish a specific data browser or editor for one or more tables that are linked via relations. We generate a system for linked screens which ensures direct movement from one screen to another.

The primary objects in our system are entities and relations among them. We have defined various screen types with various user interfaces and various levels of functionality for each entity or relation. On the screen, for instance, we can represent an entity, links to other entities (relations in the ER model), information about linked entities, or we can represent the entities that are linked by several relations. Screens are generated for each entity in accordance with its type and for each relation if the rule of the screen generator do not dictate different behavior.

Unlike a common DBMS, Simple Database Browser defines various semantically logical views of data. These views merge many data units on the basis of their entity and relation types. Such browsers are useful in reviewing the content of a database, and it can be used by system designers, maintainers and administrators.

The principle of operating of the simple database browser is shown at Fig. 6. The template *Entity view extension with links* are applied to entity Company. This template displays all of the entities which are directly linked through relations with the specific entity (with entity Company in our example). View options states what attributes are displayed.

### C. The Logical Database Browser

At the level of ideas, it is not important whether the data model describes a physical or a logical database.

The Logical Database Browser is another type of UDB, and it hides the physical implementation of the database, allowing users to review only that information which they are authorized to see.
We put together a data model that includes only the necessary entities, attributes, values and relationships, grouped and linked in the desirable way. In fact, this establishes the ontology of the requested part of the database in a way, which the end user understands. Data querying requires services that can deliver data from their source, as well as wrappers, which can translate the delivered data in accordance with the logical data model. The principle of operating of the logical database browser is shown at Fig. 7.

D. The Information Systems UDB

The Information Systems UDB (IS UDB) is a type of UDB that is essentially based on the same principles as the Logical Database Browser. What is new in the case of the IS UDB is the resources which allow it to link objects from many different information systems. This means that the ontology of the entities, which are used by systems, is established and used. The use of the IS UDB in integrating a country’s information systems is described at [1][2][4].

An IS UDB is a tool, which enables the selection and browsing of information from information systems [3]. An IS UDB has several advantages:

- It allows for a review of linked data;
- It allows for the selection of data on the basis of full use of the limited opportunities that are afforded by the specific system;
- It allows for simple addition of new data sources;

- It allows for simple structural and semantic changes in existing data sources.

The concept of a UB is presented in Fig. 8. The operations of an IS UDB are based on interpretation of meta-models. Local Logical Object Models describe the data model of each data source and the ability to extract information (functions). Global Object Model describes the global classes of entities which links related entities from the various data sources. An IS UDB can request data from data sources. The user can find the necessary information in the data source, using keywords or by looking for objects that are linked to the desired object, browsing through links for this purpose.

Data sources are various information systems usually accessed through pre-defined access functions. Data sources are usually based on various technologies, and the access functions are established in different environments and through different technologies. Data in various data sources tend to be heterogeneous. This heterogeneity can be manifested at various levels, beginning with different database platforms and ending with a situation in which the same data have different data models in different data sources.

An IS UDB contains a unified, global and conceptual meta-model of data sources. A meta-model is a UML classes diagram with specific extensions. Meta-model is made by developer who integrates systems based on users views of data and functionality provided by data sources. The database of a meta-model contains a whole series of metadata:

- A list of data sources;
- The object classes which exist in data sources and their attributes;
- Links among object classes, including attributes which can be used for search purposes;
- Functions which can be used to obtain information, including fields that are function input and output;
• Types of class presentation – which attributes are to be shown in which format and which representation.

Two data sources and one base class are presented in Fig. 9.

Fields in square brackets are invisible fields and are used for search purposes only. A solid line with arrows means that if you know information from the entity that is at the starting point of the arrow, then you can get related information from the entity that is at the opposite end. An interrupted line shows the relationship between a normal entity and a base class entity. The base class entity is an entity, which is used to link entities from different models. Base class entities are contained into Global Object Model.

The navigational principles of the Universal Browser can be described via an example shown in Fig 9.

The IS UDB receives information from data sources via wrappers. This approach has a number of advantages:

• It enables access to data sources via different protocols and methods;
• Access to the data of data sources is usually limited, and certain stored procedures can be used to query data – the wrapper allows us to execute only authorized functions;
• Querying data sources via functions allows us to ensure the easy transfer of real data from the physical data model to the logical data model, which is more understandable for the user.
• Wrappers translate information from the specific data format of the data source to the internal data format of the IS UDB.

Information about functions is stored in the IS UDB metadata database – the defined input and output fields for each function. Each input field may be mandatory or optional.

VI. THE ARCHITECTURE AND OPERATING PRINCIPLES OF THE SEMI-STRUCTURED DATA UDB

Semi-Structured Data UDB (SSD UDB) is intended to use for retrieval, analysis, translation, storage, integration and browsing of semi-structured data. The SSD UDB is largely based on the IS UDB. The main idea is to convert semi-structured data into structured data. Many different types of ontology are used to link information. When structured data are accompanied with data search and extraction services, this creates a specific informational system with which the IS UDB can work (Fig. 10).

The source of semi-structured data is usually a Website or some type of document (pdf, doc, xls). These data sources usually do not offer data access functions, which ensure access to concrete data. Usually it is only possible to receive and analyze the entire document.
In order to ensure access functions, we propose that data from the data source be pre-selected and placed into the SSD UDB structured data database.

A. Retrieval and transforming semi-structured information into structured information

We shall review a specific class of semi-structured data. Within this class we find data, which are composed of two parts – a hierarchic classification and the data of concrete objects. These data can be partly structured. Our example is the product catalogue of a manufacturer or retailer.

Before a UDB can operate with semi-structured data, they have to be converted into structured data. In general terms, this is a problem, which has not yet been fully resolved. In our case, we are structuring only that information, which can be, structured relatively easily – classification and the structured part of the data. The unstructured part remains unstructured, and searches therein are possible only on the basis of keywords (text). In the product catalogue, which we are using as an example, it is easy to structure the product classification and to assign concrete properties to concrete products.

Classifiers of data are, in essence, examples of ontologies (classification ontologies). Data models can be established for data sources, which are essentially ontologies (product ontologies).

The structured data, which are obtained, must be stored in a database. Product classifications that are extracted from data sources are stored in the ontology of these classifiers. The main problem is that the structure of product database is never known in advance, and so it is problematic to establish the database. Therefore product data are stored in a universal structure (Fig. 11). This universal structure means that the product or its part is an entity with a specific set of attributes. Each attribute has a set of values. Entities can be linked by relations. In order to store data in the universal structure and to select them from the universal structure, we use the information about the data model, which is found in the product ontologies.

The mechanism for converting semi-structured data into structured data is seen in Fig. 12.

Each source of semi-structured data has a different format. Websites are analyzed, and wrappers are developed which can parse Web pages and extract the necessary information. If it is a page with product classification, then the classification is extracted. If the page contains product information, then the data of the concrete product are extracted.

A Website usually has a main page from which various links can be used to browse through the entire catalogue – pages with classifications, as well as pages with descriptions of concrete products.

Before a data source can be analyzed and data can be collected, it is necessary to analyze the pages and to establish ontology for the local products of the source. Also, a wrapper, which can parse the concrete data source, must be established.

For making the rounds, there is an engine for filling in and renewing data, which looks at the necessary pages on the Website and uses the wrapper to extract the necessary information and then to store it locally. In parsing the product classification, for instance, this engine renews and supplements the local and physical classification ontology. The filling in and renewing of data is handled by the engine through information about the model of the data source that is stored in the product ontology. Then it can record precise data in the product database.

B. Searching for and browsing semi-structured data

Data searching and browsing from the structured database is illustrated in Fig. 13. It is a modified version of the IS UDB.
The operations of the engine and the presentation tool are almost the same as is the case with the IS UDB. The engine through the use of a wrapper extracts data from the database, and then the engine transfers the extracted data to the presentation tool, which then presents the data to the user. The presentation tool then transfers the user input to the engine. The only other thing that needs to be said is that if the IS UDB engine used only one metadata base, then the engine in of SSD UDB will use information from several classification ontologies and product ontologies for the data search and browsing.

The structured database serves as a data source for the IS UDB. The data access wrapper allows the engine to access information that has been stored in the structured database. We do not have any specific access functions that would allow us to access a structured database, and so the wrapper interprets data from the product ontology so as to generate dynamic requests with select data from the structured database.

VII. CONCLUSION

In this article we have reviewed the development of our Semi-Structured Data Universal Data Browser (SSD UDB). Initially it was a simple mean (Simple Database Browser) for viewing a specific database, one that was of use only to software developers who needed to review information in a database at the level of tables, fields and relations. The Logical Database Browser was developed by introducing various ontologies or logical views of data from a database or information system, thus establishing a browser that is of use to the end user. There was then further development in the form of integrating information systems - information from registers of national importance was consolidated (IS UDB).

At the current level of UDB development, we are looking at the possibility of integrating information from many semi-structured data sources (Web pages). For experimental purposes, we have chosen Web resources, which offer information about the products of various manufacturers.

We transform the extracted data and store them in a database, which is based on a meta-model. In order to maintain the logic of the Web resource, we are using several physical ontologies, which describe data sources. Integration among the many sources is ensured by global ontologies. For purposes of convenience and flexibility, each data source also involves logical ontologies, which are an intermediate step between physical and global ontology.

Our next step is to check SSD UDB operations in relation to the selected class of tasks. Our approach is based on the fact that we turn semi-structured data sources into structured data sources via data extraction services, thus creating a situation in which the application of the IS UDB has already been tested. We will take a separate look at how the search and display of information can be made more convenient for the user. That will largely depend on the many ontologies that are developed so as to describe the world view of various users, along with their habits related to working with information systems.

One additional direction of our further work is research of virtual ontologies. Virtual ontologies can be established dynamically, because the global level ensures the linkage of ontologies that exist at far lower levels on the basis of various aspects. For instance, a potential method, which should be tested in practice, is the merger of many physical ontologies into a single ontology. Similar ontology nodes are brought together, and all names are left in place so that they can be identified. This basically produces a graph, and users must be given a way to maneuver through it easily so as to find the appropriate node.

REFERENCES


