Semantic Web-driven Agent-based Ecosystem for Linked Data and Services

Oleksiy Khriyenko
Industrial Ontologies Group, MIT Department
University of Jyväskylä, P.O. Box 35(Agora)
Jyväskylä, Finland
oleksiy.khriyenko@jyu.fi

Michal Nagy
Industrial Ontologies Group, MIT Department
University of Jyväskylä, P.O. Box 35(Agora)
Jyväskylä, Finland
michal.nagy@jyu.fi

Abstract **— We are surrounded by data – data about events, our daily activities, a multitude of products and services from different vendors, etc. This data is playing a central role in our lives. It helps us make better decisions. Increasing numbers of individuals and organizations are contributing to this huge flow by sharing their data with others, including Web-native companies (such as Google, Amazon, Facebook, YouTube, Twitter, etc.), newspapers, public governmental bodies, various research initiatives, etc. In turn, third parties are consuming this data to build new businesses, provide new services and accelerate scientific progress. However, very often new service creation has an obstacle – limited data availability. Becoming accessible later, data may cause reimplementation of the service that might cost too much and user will be left without improvement of the service. In this paper we combine the existing technologies, highlight the challenges and show the way that might help solve the problem. In order to elaborate Semantic Web-driven Agent-based Ecosystem for Linked Data and Services we utilize the so-called UBIWARE platform. UBIWARE is a semantic middleware platform for Ubiquitous Computing.**

Keywords – linked data, semantic service ecosystem, agent technology, service infrastructure, semantic service integration

I. INTRODUCTION

In the beginning of the computing era, simplicity of programming languages allowed people to write programs on paper. However, the complexity and size of programs has grown. Also, large programs were thought of as complex interconnected graphs in comparison with the linear structure of text document. As a result we had to make more efforts to express the full program as a collection of text files. Programs were separated to different files depending on the nature of data and role it played. Previously, programs were concentrated on local data, which was used from the memory of the computers that were running it. The majority of programs today have to do both: persist data and connect to remote data that is managed by another party.

In order to increase their competitiveness, many organizations are looking for a way to enhance their internal data with external information. However, the integration of information from different heterogeneous systems is challenging. In order to achieve it, we have to concentrate our efforts not only on an internal system integration framework, but also on a common approach towards distributed collaborative environment of heterogeneous components. The main trend of programming today is that most new system will not only need to persist their data, but will also need to connect to other programs across the Internet. Indeed, we often do not know which data will be local or which will be remote. Thus, now it is an appropriate time for systems and services to treat all data references as potentially pointing to data that resides somewhere else on the Internet. By taking this homogeneous approach, programmers can focus on the basic logic of what needs to happen rather than care about the data. Building connections between systems currently requires extra work that has to be repeated for each connection. This is additional work from a technical and financial point of view. Rather than building huge data storages, we should think about data as a complex graph that is connected across and distributed over many systems. The service oriented architecture (SOA) [1] approach to connecting existing data sources is an important step towards making it easier for existing systems to communicate with each other. However, again, in contemporary programming languages, the SOA approach has to be managed explicitly by the programmer. If we adopt programming paradigms that treat all data as being data on another system, we free the programmer of certain routine tasks and giving him/her more time to concentrate on other aspects of the system. Instead of huge data storages we should think about networks of data.

To achieve the vision of ubiquitous Web, the next generation of integration systems will need different methods and techniques such as Semantic Web [2] [3], Web Services [4] [5], Agent Technologies [6] and Mobility [7] [8]. Semantic technologies are viewed today as a key technology to resolve the problems of interoperability and integration within the heterogeneous world of ubiquitously interconnected objects and systems. It is evident that for two systems to communicate with each other, they have to use a standard language that they can both understand and share a standard ontology. There are different points of view concerning the uniqueness of the ontology. Some think that “one ontology approach” is the best possible solution to have one common standard and avoid ambiguity. Others consider this as an illusion and, rather than dreaming of agreeing on one common ontology, they suggest always working on the assumption that our systems should handle interactions with other systems using multiple different ontologies even within a single domain. It is one more aspect that increases the

 ISBN: 978-1-61208-152-6

complexity of data management and distributed system interoperability.

The rise of software development methodologies further separated the way that programs and data are managed. There are many modern software systems with a lot of configuration options that can be user customized. It removes a clear boundary between the programming and pure data management. Nowadays, users want even greater control over how their computers manage their data and would like to become programmers on at least a higher system level. Users would like to manage not only the data, but program a system behavior through its configuration.

All the mentioned difficulties and complexity of the upcoming Web (web of autonomous intelligent systems and web of data) tell us about the necessity to elaborate the correspondent ecosystem with the appropriate tools and capabilities to support each player in the Web to operate there in a proper way. The Ecosystem should hide all the complexity and control the technical part of interoperability and unambiguousness. This paper presents a possible way of ecosystem elaboration based on agent-driven infrastructure for Linked Data and Services interoperability.

In the second section, we propose an architecture that would simplify the creation of ubiquitous services. We stress the importance of the Linked Data approach as an important enabler of service interoperability. The third section describes the infrastructure of such an ecosystem based on Multi-Agent technologies. In the fourth section we address several challenges of the Linked Data infrastructure. The fifth section concludes the article.

II. Ecosystem for Linked Data and Services

A. *Towards Data-free Ubiquitous Services*

In today's world we can find different kinds of data in different forms. In general this data can be utilized by third parties in order to provide additional services. However, very often new service creation faces an obstacle in the form of data unavailability. Usually, ignorance of certain data availability and accessibility limits us to develop one or another useful service. If some data becomes available and accessible later and some existing service would like to utilize it, it may cause a service reimplementation. This reimplementation might cost the service provider too much in terms of time and money. Eventually, the user might be left out without any service improvement. To avoid such a case and allow new services to be developed in a more flexible way, we have to consider Semantic Web-based infrastructure for linked data and services. Standardized approaches like Semantic Web technology and Ontology-based development may help us to develop services, which are not bound to data, but operate with data on the semantic level. Following this approach, the service logic can be independent of the particular data source availability and still provide a service to the customer. Such a service elaboration approach and semantic organization of virtual data source makes the servicing context-aware, less sensitive to the unavailability of data and open/extendable for data that might be accessible in the future. Thus, one of the main parts of service infrastructure is common shared virtual data source – Semantic Data Space of linked information. Fig. 1 shows a general architecture of manageable Linked Data infrastructure for services and applications.

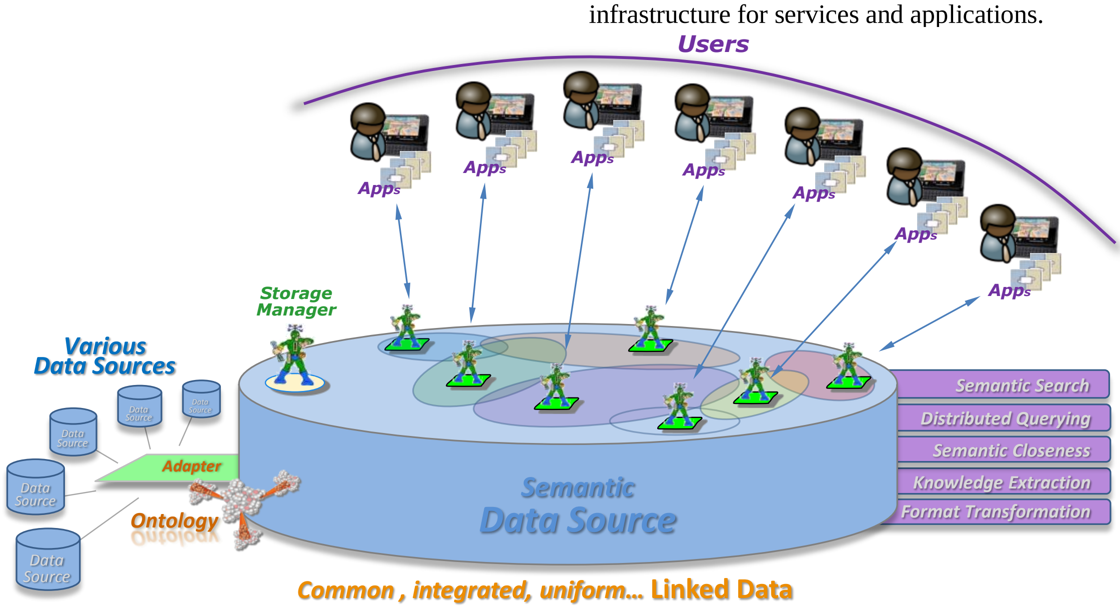

Figure 1. Semantic Data Space of linked information.

This smart data storage is an intelligent mashup of different heterogeneous sources of information with a dynamic private space for each user. It might be a source of any public or private information with correspondent access control mechanism. Depending on the user profile, activity performed by him/her and contextual information (e.g. location), the smart data storage should create the correspondent information space for services and applications used by user for the mentioned activity. In other words, this space will contain only the activity- and context-related linked data. Each information space has a manager that manages the data, adds relevant data and removes irrelevant data from the space. Such smart semantic organization of the common data storage allows services/applications to be automatically switched between different real data sources on-the-fly depending on the context. Such data organization allows creation of mashups through flexible semantic orchestration and user-driven choreography of services and applications. Such an approach very well supports and complements the new strategy of Nokia towards Internet-based mobile application solutions and the mixed reality concept. Following this approach, applications will get access to application independent, but contextually relevant, information. Such architecture allows us to move the data accessing and data processing part to the application independent layer and perform a reliable and trusted data access control.

For example, let us consider some notification service in the user's mobile phone, where the user has specified his/her interest in "yacht exhibition" events. The user is not going to visit some specific exhibition that might take place somewhere far away, but he/she would be interested in visiting an exhibition close to his/her current location. In this case, the contextual information is the user's location and the type of relevant events. Based on this context, the ecosystem creates a semantic data space with relevant information published by other systems/services (exhibition centers, yacht clubs, city and region event centers, etc.). The application in the mobile phone is developed in a general way and does not care about the data. It just sends the appropriate pattern of the request and listens to the response from the smart data space manager, which performs all search and matching processes. Even in the case when a certain exhibition has a restriction and is open only for the members of some yacht club, the application does not care about this fact. The correspondent data space manager itself checks this information based on the user profile and available linked data about the yacht club members. Thus, the complete semantic data space takes the responsibility for linked data management and allows applications and services to become ubiquitous and data independent.

B. Linked Data as a Basis for Service Interoperability

Unstructured data, heterogeneity of different data sources and many other problems become a bottleneck of automated data integration, processing and reuse. To make data ready to be processed by external intelligent algorithms and methods, data sources and data should pass through the semantic adaptation. If we are aiming at making a step towards intelligent data processing, to intelligent use of information from different data sources, to intelligent management of huge data storages, to knowledge extracting from huge archives of data, etc., we have to make the data linked. The data should be accessible in common uniform way through one virtual linked semantic data source, even if originally it is located in different data sources.

From the early beginning, the World Wide Web (WWW) was a system of interlinked hypertext documents accessed via the Internet. The Web allows document creators to freely choose whether they will or will not refer to any other document. As a result we have got a huge mass of information that was managed by search engines and browsers. However, with rapidly growing amount of information on the Web, the society needed some advanced mechanisms for data management. Later on, Semantic Web technology was announced as a "web of data" that enables machines to understand the semantics (meaning) of the information on the WWW. It extends the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are related to each other. This enabled automated agents to access the Web more intelligently and perform tasks on behalf of the users. It was the first attempt to arrange and standardize the data and data management.

Due to huge amount of areas and aspects that Semantic Web technology tried to cover, the community started to elaborate different standards and techniques to solve different problems. As a result, we have a big variety of separated islands of information and management systems. These information islands internally follow the Semantic Web vision, but are heterogeneous from the general (global) interoperability point of view. This leads to the fact, that society and especially its business-oriented part started to doubt that such widely spread activity will be so much beneficial for them. Only some applications and systems in restricted domains became really useful. Most probably, the reason for this is the decentralization of uncontrolled activities, which creates new problems and bottlenecks on the way towards ubiquitous Semantic Web.

It was evident that sooner or later the concepts and ideas of Semantic Web will be reformulated and presented in a simpler way to show a small but important step of technology applicability. As we see nowadays, the Linked Data concept, introduced and promoted by the fathers of WWW and Semantic Web [9] [10], becomes popular. The concept describes a method of publishing structured data, so that it can be interlinked and become more useful. Just as the WWW has revolutionized the way we connect and consume documents, Linked Data can revolutionize the way we discover, access, integrate and use data. It is built upon the Web as the ideal medium with ubiquity, distributed and scalable nature, mature and well-understood technology stack. There are no doubts that Semantic Web is a very promising technology of the future. It definitely lacks more centralized management or at least an environment that plays coordinative and supportive role and directs users to the proper utilization of the technology.

According to Tim Berners-Lee's "5 stars" advice to enable Linked Data [9], the principles include: making data available on the Web (in whatever non-proprietary format) as machine-readable structured data, utilizing open standards from W3C (RDF [11] and SPARQL [12]) to identify things and finally linking the data to other people's data to provide the context. But, it seems that it is not enough to define only requirements or principles. To facilitate proper utilization of the technology and increase the benefits of it, it is reasonable also to provide technical support in a form of Semantic Web oriented ecosystem platform with appropriate tools and services that do all the “dirty work” and keep the whole ecosystem in proper condition.

III. AGENT-BASED ECOSYSTEM INFRASTRUCTURE

Semantic Data Space is a complex data management system based on autonomous data space managers, distributed original data sources and supportive ecosystem tools and capabilities. In recent years, complexity of computing environments has grown beyond the limits of human system administrators’ management capabilities. With the advent of service-oriented computing (SOC), computing environments have become open and distributed, and components are no longer under a single organization control. Autonomic computing systems are expected to free system administrators to focus on higher-level goals [13]. Self-configuration, self-healing and self-optimization can be performed by autonomic computing systems without human intervention. As such, autonomic computing systems strongly resemble multi-agent systems (MAS). MAS, in turn, interact with services, as designed and developed within SOC. To make the system intelligent, dynamic and autonomous, we have to utilize Agent Technologies [14]. From the implementation point of view, agents are the next step in the evolution of software engineering approaches and programming languages. It is a step following the trend towards increasing degrees of localization and encapsulation in the basic building blocks of the programming models [15].

Developing and maintaining large-scale, distributed applications is a complex task. Middleware has traditionally been used to simplify the application development by hiding low-level details and by offering generic services that can be reused and configured by application developers. However, the middleware technology has not been keeping up with the growing demands that emerge in the digital society. The scale of distributed applications is rapidly increasing. The range of users that compose and configure applications has expanded significantly, and the increased scope of distributed applications has also resulted in more advanced application composition scenarios.

A. UBIWARE Platform: Integration Infrastructure for Heterogeneous Distributed Components

We base our research towards the elaboration of Semantic Web – driven Ecosystem for Linked Data and Services on UBIWARE platform that follows the GUN vision [16]. The Platform is a development framework for creating multi-agent systems. It is built on top of the Java Agent Development Framework JADE [17], a Java implementation of IEEE FIPA specifications. The name of the platform comes from the name of the research project, where it was developed. The UBIWARE project [18] introduced a new paradigm in software engineering and elaborated an approach towards the creation of semantically enhanced agent-based integration middleware that makes heterogeneous resources proactive, goal-driven and able to interoperate with each other in collaborative environment. In this project, a multi-agent system was seen, first of all, as a middleware providing interoperability of heterogeneous resources and making them proactive and “smart”.

The core of the platform gives every resource a possibility to be smart by connecting a software agent to it. This agent enables the component to proactively sense, monitor and control its own state and communicate with other components. The core component of the UBIWARE platform is a UBIWARE agent depicted in Fig. 2. It can be seen as consisting of three layers: the Behavior Engine implemented in Java, a declarative middle-layer (Behavior Models corresponding to different roles the agent plays), and a set of sensors and actuators, which are again Java components. We will refer to these as Reusable Atomic Behaviors (RABs). The middle layer is the beliefs storage presented in *Semantic Agent Programming Language (S-APL)* [19], which is a Resource Description Framework (RDF) [11] - based language. S-APL integrates features of agent programming languages (like AgentSpeak [20] and AFAPL [21]), semantic reasoners (like CWM), querying languages (like SPARQL [12]) and agent communication content languages (as FIPA SL [22]).

The main goal of the Platform is to provide interoperability between heterogeneous resources (applications and systems in our case). This can be achieved by semantic adaptation and by assigning a proactive agent to each of the resources. The communication, resource discovery and resource usage are performed via the correspondent agent. The Platform has inter-platform communication mechanisms and allows integration, orchestration and choreography of resources registered and located in different platforms. UBIWARE is not an application like an operating system, word processing software or Internet browser. It is a set of tools that helps people develop software. With respect to Cloud-based integration environment interoperability, we consider the UBIWARE platform a tool that allows automatic discovery, orchestration, choreography, invocation and execution of different Business Intelligence services.

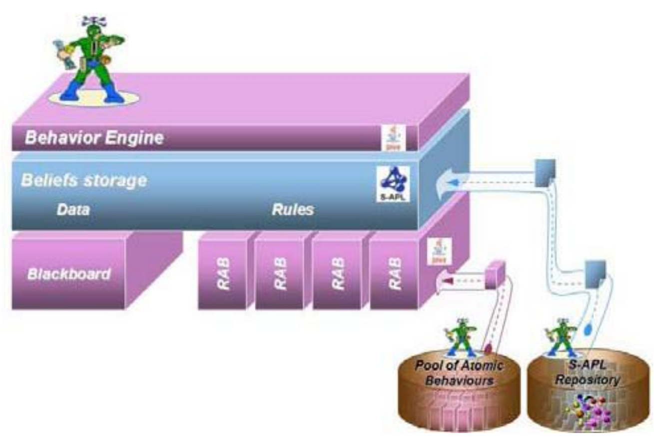

Figure 2. UBIWARE Agent architecture.

B. Semantic Data Space as a complex data management system

The UBIWARE platform-based ecosystem assumes that we have a network of interconnected platforms that can be extended any time it is needed. As it was mentioned in the previous section, the core entity of the platform is an agent. Each resource (data source, service, human, etc.) has an agent assigned to it. The agent represents its resource and performs all the communications with other resources (resource agents) in the ecosystem.

Users, data source adapters and other applications manually, semi-automatically or automatically create and edit the Web of linked data. The first layer of the ecosystem (see Fig. 3) is populated by Platforms that help to create, store and manage this data. On the same layer, the ecosystem has supporting tools, interfaces and other infrastructure services (browsers, search engines, similarity measurement modules, various supportive registries and systems) to be utilized by Platforms' managers and users. Thus, this layer represents a network of linked data.

Another layer is represented by agents. These agents are managers of personal data spaces. They form a smart semantic data space of the ecosystem. Such personal data spaces are generated each time when some application or service starts to consume linked data. These managers are responsible for proper context-dependent data access control and for relevance and unambiguousness of provided data. In other words, based on contextual data provided by applications/services, they on-the-fly create and manage datasets that perfectly match correspondent patterns provided by applications. During the whole life-cycle of the data space, the manager agent updates, adds, deletes the correspondent dataset with respect to the changes of the contextual information (user profile, location, tasks and conditions, etc.). To enable this, the manager communicates with the application and with other platform managers. It also utilizes infrastructure services to browse, search and filter linked data.

C. Capabilities and Tools of the Linked Data Infrastructure

The main functionality and main purpose of the proposed ecosystem is to provide infrastructure and tools that facilitate the process of Linked Data creation, browsing, search and access. Such infrastructure can be considered a perfect playground for semantic-driven applications and services that are given ubiquitous access to the complete data space.

A key factor in the re-usability of data is the extent to which it is well structured. The more regular and well-defined the data structure is, the easier it is to create tools to reliably process it for reuse. While most Web sites have some degree of structure, the language in which they are created, HTML, is oriented towards structuring textual documents rather than data. As data is intertwined with the surrounding text, it is hard for software applications to extract snippets of structured data from HTML pages. There were attempts to use various microformats to embed data to HTML, but all of them are restricted with a small set of types of entities and attributes and are not suitable for sharing arbitrary data on the Web. A more generic approach to make structured data available on the Web is Web API usage. Web APIs provide simple query access to structured data over the HTTP protocol. The advent of Web APIs has led to an explosion in small, specialized applications (or mashups) that combine data from several sources, which are accessed through different APIs. While the benefits of programmatic access to structured data are indisputable, the existence of a specialized API for each data set requires significant efforts to integrate each novel data set into an application. Every programmer must understand the methods available to retrieve the data from each API, and write custom code for accessing data from each data source. Thus, Web APIs make data accessible on the Web, but they do not place it truly on the Web, making it linkable and therefore discoverable.

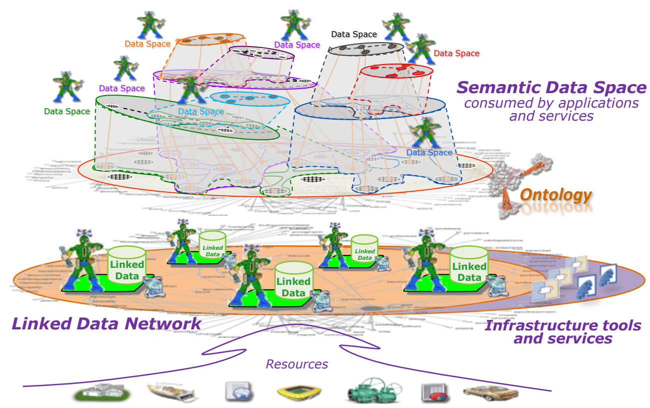

Figure 3. Semantic Data Space – a complex data management system.

The GUI of the Platform should help the user create Linked Data in the appropriate form. This controllable way of Linked Data creation minimizes the user's efforts and hides all technical complexity of the process. Such GUI has to be smart and configurable. This allows the user to connect and utilize different domain specific ontologies, and provide context dependent guidance for the Linked Data creator. At the same time, together with simple static resource transformation, we have to consider more complex resources and systems that produce dynamic data. There are a lot of huge data storages (that cannot be managed manually) and systems that produce an avalanche of data. This data is very often available only in human readable form. Within the UBIWARE platform we elaborated a resource adaptation framework and tools that facilitate creation of semantic adapters for various resources. These tools can automate the adaptation process for huge and dynamic data sources. It is possible to automate the process of Linked Data creation via correspondent data adapters. As a proof of this approach, in the iSCOPE project we adapted several web pages and services that dynamically publish event data. Then, we semantically annotated this data and made it available to remote mobile applications.

Nowadays, the human becomes a dynamic and proactive player in a large heterogeneous distributed environment with a huge amount of various data, services, devices, etc. Therefore it is necessary to provide a technology and tools for easy and handy human information access and

 ISBN: 978-1-61208-152-6

manipulation. If the Web of Data is based on standards and a common data model, it will become possible to implement generic applications that operate over the complete data space: Linked Data browsers (which enable the user to view data from different data sources), Linked Data Search engines (that crawl the Web of Data and provide sophisticated query capabilities on top of the complete data space). These two general types of applications should be considered as general infrastructure tools/capabilities and used by all other applications as services in the Linked Data ecosystem. But, such functionality should not only be available for humans. The Ecosystem infrastructure should provide the same capabilities for machines or applications as well. This means that the Platform should be equipped not only with GUIs and tools for human operation on Linked Data, but also equipped with APIs, correspondent GUIs and utilities for application and service developers.

Another important aspect related to data browsing is context-sensitive visualization of data to a user [23] [24]. Context-awareness and intelligence of the user interface bring a new feature that gives the user a possibility to not get just raw data, but also required information based on a specified context. A user needs a fast and convenient way to specify what he/she is looking for and get the semantically closest resources to his/her query. Resource closeness/similarity search is one of the most useful features that users need during the information retrieving process [28]. The similarity search has become a fundamental computational task in many applications. Thus, intelligent visualization of Linked Data and context-aware filtering of relevant data becomes a very important functionality of the GUI and Linked Data browsers.

IV. CHALLENGES OF LINKED DATA INFRASTRUCTURE

A. *URI aliases vs. Semantic Web ecosystem control*

One of the bottlenecks of Semantic Web is the huge amount of URI aliases, the multitude of URIs in different namespaces identifying the same entity. There is a common agreement saying that two URIs referring to the same resource should be connected using a link of type "http://www.w3.org/2001/07/owl#sameAs". However, due to lack of control, these links are not used often. As a result, in many cases there are several unlinked information sources that describe the same resource. At the same time, even if the "owl#sameAs" link is used, it does not mean that it refers to the original resource or even the same resource. These resources are same just form a particular publisher's point of view. It is another weak point of URI aliases.

To achieve unambiguousness, each resource should have only one URI, defined by the resource owner that takes care of the resource: a person, organization, community, or any other correspondent authority. The Semantic Web ecosystem platform should provide a mechanism for resource creation and correspondent tools that autonomously browse the Web and search for similar resource definitions to initiate a process of ownership detection. The same similarity search engine can be used for the required resource URI search to make the proper reference. Thus, the main information about the resource is provided by the owner (responsible authority). Data can be updated by the owner on request from third parties with correspondent support from the ecosystem platform. At the same time, we cannot deny the possibility for others to have their own opinion and to provide additional information (define new values for the resource properties) about a resource they do not own. However, the owner may ignore this information. In this case, any additional data linked to the resource by a third party should be marked by the ecosystem platform as unverified information and the correspondent owner will be notified about it. All such unverified information should be stored by the ecosystem platform under the correspondent contextual description [25] and be available under these contexts for other users.

Thus, we still allow different views and opinions to be expressed and, at the same time, we avoid multiple URI aliases. We do not need to create a centralized naming authority to assign URIs that would introduce administrative and bureaucratic overhead. Instead of it, the Semantic Web ecosystem will automatically search for similar resources and apply correspondent actions in case of data duplication. Someone might think that this approach can be a case of "central point of failure", but it is not. Data is still located in different data sources and, if data warehousing techniques (replication, redundancy of data, etc.) would not help and the main data would be lost, then later on it can be regenerated from distributed locations. Otherwise, if we continue to use aliases and apply hundreds of millions of "owl#sameAs" to express identity links, we will have a huge unmanaged mass of claims of different parties rather than facts and related set of additional claims in correspondent contexts.

B. *URI's dereference*

The Web is intended to be an information space that may be used by humans as well as machines. There are two different strategies to make URIs that identify real-world objects dereferenceable. Both strategies ensure that objects and documents that describe them are not confused. They also ensure that both humans and machines can retrieve appropriate representations. The strategies are called 303 URIs and hash URIs [26]. The hash URIs method has the advantage of reducing the number of necessary HTTP round-trips, which in turn reduces access latency. Then again, the descriptions of all resources sharing the same non-fragment URI part are always returned to the client together. This leads to large amounts of data being unnecessarily transmitted to the client. The 303 URIs method, on the other hand, is very flexible because the redirection target can be configured separately for each resource. There could be one descriptive document for each resource or one large document for all of them.

In our opinion, it is more reasonable to leave resource URI references in a machine-readable form, especially because we are going to utilize data browsers (machines in this case) to present information to the users. Moreover, we should not limit ourselves to just one human readable representation of the resource. We have to consider different representation forms and views to present data depending on different contexts. Thus, a URI points to an RDF description of the resource. This description contains a set of context dependent statements with a property (for example "#visualRepresentation") that refers to correspondent human readable/viewable resource representation. Now, when the Linked Data browser reaches the resource RDF description, it chooses (depending on the context) the appropriate representation form. After that it shows the resource to the user directly or through appropriate visualization services. Some of the techniques relevant to context-aware information visualization and browsing techniques can be found in our previous works [23] [24] [27] [28].

C. Incoming links: symmetry of the properties

In most cases data is linked in one direction. An RDF triple links one resource to another. Usually, describing a certain resource, only relevant (from the point of view of this resource) resources are linked through correspondent properties. It allows us to browse and discover those linked resources. But what about discovering the original resource? While linking another resource to the original one, we have to ensure that if other resources do not have back links to the original one, then at least the ecosystem collects this information under the correspondent context and makes it available for applications and services. It makes our original resources discoverable from the descriptions of other resources additionally created by the ecosystem. Such incoming links enable the user to navigate backwards with Linked Data browsers. In addition to that, they enable crawlers of Linked Data search engines to discover original resources and continue crawling. Naturally, the "owners" of the resources should be notified by the ecosystem platform manager about additional automatically created descriptions. Later, if the "owner" wants, such description can be added to the main resource description with the support of correspondent ecosystem tools.

D. Context-sensitive equivalence of ontologies

Another challenge for Linked Data comes from ontologies, more precisely, from the multitude of them created for different domains. Even if the ontologies are defined for different domains, it might happen that some of the properties have the same meaning and users might describe the data differently following different ontologies. In some cases, such equality of the properties might be context dependent and should be treated with respect to contextual conditions. To manage this heterogeneity, the supportive ecosystem should have a registry that contains a context-sensitive definition [25] of the properties' similarity. The process of similarity detection could be done in a semi-automated way by a similarity search engine with an approval from the responsible expert or/and with a help of weighted feedback from the users. Thus, supportive tools of the ecosystem (browser, search engine, etc.) could automatically request such a registry to get the correspondent list of similar properties and provide a better service.

E. Context –aware Policy-based Data Access Control and Similarity-based Data Search

In order to elaborate the mechanisms for context-aware policy-based resource access and contextually related information retrieving, we require a model. This model should present the influence of the contextual information on data search and retrieving process, on the level of relevance of the links and similarity of the resources. Depending on the context, properties become more or less relevant. This gives us different vectors of weights. In the same way, the context might influence data access, data privacy and security issues. Thus, context-dependent policy-based control seems to be a very promising approach, which is able to keep data links flexible, dynamic and controlled at the same time. This approach should allow us not to program a fixed and hardcoded data access control and search system, but to build it with the ability to change the internal structure on the fly when the context is changed [29] [30].

In our opinion, it would be reasonable to extend traditional explicit semantic links within Linked Data with the implicit ones, for example those, which could be automatically derived by various reasoners. Among those, special attention should be paid to the "semantic similarity" links. Usually, when one queries data, one looks for the resource(s), which are "the same" as the one specified in the query. However, often none of them are found. In many cases it makes sense to find a resource "similar" to the target one. Similarity search was always a big issue within many disciplines and it is especially important for Linked Data. Similarity search should also simplify extensive work aimed at recognizing same resources that have different URIs (see subsection A). Usually we see first that some resources look similar and therefore in practice could be the same ones and then check on the identity of the resources. Therefore explicit similarity links between data entities could be discovered as a result of appropriate similarity search procedures. The challenge here is that some resources being very different in one particular context could be considered similar ones in some other context. As one of the results of the UBIWARE project we developed a prototype of a context-sensitive visual resource browser [27] that we use as a basis for the Linked Data browser and similarity search engine.

V. CONCLUSIONS

The Linked Data concept provides us with the possibility to create a complete data space for humans, applications and services. Linked Data is essential to actually interconnect the Semantic Web. The paper presents an agent-based ecosystem with the Linked Data infrastructure as a playground for Semantic Web-driven services and applications. Within this paper we reviewed relevant technologies, showed the benefits and highlighted some challenges of the Linked Data infrastructure with possible ways that might provide the solutions. To facilitate proper utilization of the technology and increase the benefits of it, we need technical support in a form of an ecosystem platform with appropriate tools and services that do all the "dirty work" and keep the whole ecosystem in the proper condition. Middleware has

 ISBN: 978-1-61208-152-6

traditionally been used to simplify application development by hiding low-level details and by offering generic services that can be reused and configured by application developers. To build such an ecosystem we base our research and development on the UBIWARE platform. The UBIWARE platform is a tool for proactive interoperability of distributed heterogeneous components. Utilizing Semantic Web and Multi-Agent System approaches, the UBIWARE platform makes the Ecosystem proactive and flexible. The paper describes capabilities and tools of the proposed ecosystem.

ACKNOWLEDGMENT

This research is based on activities of the Industrial Ontologies Group in the latest projects in Agora Center and Department of Mathematical Information Technology (University of Jyväskylä, Finland): UBIWARE, iSCOPE and projects under Cloud Software program in TIVIT SHOK. The authors are very grateful to the members of IOG for their fruitful cooperation within this research topic.

REFERENCES

[1] J. Domingue, D. Fensel, and R. González-Cabero, “SOA4All, Enabling the SOA Revolution on a World Wide Scale”, In: Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, California, USA, IEEE CS Press, 2008, pp. 530-537.

[2] Semantic Web, 2001. URL: http://www.w3.org/2001/sw/, Accessed June 15th 2011

[3] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web”, Scientific American 284(5), 2001, pp. 34-43.

[4] A. Ankolekar, M. Burstein, J.R. Hobbs, O. Lassila, D.L. Martin, D. McDermott, S.A. McIlraith, S. Narayanan, M. Paolucci, T.R. Payne, and K. Sycara, “DAML-S: Web Service Description for the Semantic Web”, 2002. URL: http://www-2.cs.cmu.edu/~terryp/Pubs/ ISWC2002-DAMLS.pdf, Accessed June 15th 2011

[5] M. Paolucci, T. Kawamura, T.R. Payne and K. Sycara, “Importing the Semantic Web in UDDI”, 2002. URL:http://www-2.cs.cmu.edu/~softagents/papers /Essw.pdf, Accessed June 15th 2011

[6] FIPA, “FIPA Interaction Protocol Library Specification Specification”, FIPA00025, 2001. URL: http://www.fipa.org/specs/fipa00025/, Accessed June 15th 2011

[7] F. Curbera, M. Dufler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, “Unraveling the Web Services Web: An introduction to SOAP, WSDL and UDDI”, Internet computing, 2002, pp. 86-93

[8] J. Clabby, “Web Services Executive Summary”, 2002. URL: http://www-106.ibm.com/ developerworks/webservices/library/ws-gotcha/?dwzone= webservices, Accessed June 15th 2011

[9] T. Berners-Lee, “Linked Data - Design Issues”. 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html, Accessed June 15th 2011

[10] T. Heath and C. Bizer, “Linked Data: Evolving the Web into a Global Data Space” (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool. 2011

[11] W3C, ”SPARQL Protocol and RDF Query Language”, 2008. URL: http://www.w3.org/TR/rdf-sparql-query/, Accessed June 15th 2011

[12] W3C, ”Resource Description Framework”, 2004. URL: http://www.w3.org/RDF/, Accessed June 15th 2011

[13] F.M.T.Brazier, J.O. Kephart, H. Parunak, and M.N. Huhns, "Agents and Service-Oriented Computing for Autonomic Computing: A Research Agenda," IEEE Internet Computing, vol. 13, no. 3, May/June 2009, doi:10.1109/MIC.2009.51, pp. 82-87

[14] N. Jennings, “An agent-based approach for building complex software systems”. Communications of the ACM 44, 4, 2001, pp. 35-41

[15] N. Jennings, “On agent-based software engineering”, Artificial Intelligence 117(2), 2000, pp. 277–296

[16] O. Kaykova, O. Khriyenko, D. Kovtun, A. Naumenko, V. Terziyan, and A. Zharko, ”General Adaption Framework: Enabling Interoperability for Industrial Web Resources”, In: International Journal on Semantic Web and Information Systems, Idea Group, ISSN: 1552-6283, Vol. 1, No. 3, July-September 2005, pp. 31-63.

[17] JADE agent platform. URL: http://jade.tilab.com/, Accessed June 15th 2011

[18] UBIWARE agent platform. URL: http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE_details.htm, Accessed June 15th 2011

[19] A. Katasonov and V. Terziyan, “SmartResource Platform and Semantic Agent Programming Language (S-APL)”, In: P. Petta et al. (Eds.), Proceedings of the 5-th German Conference on Multi-Agent System Technologies (MATES’07), 24-26 September, 2007, Leipzig, Germany, Springer, LNAI 4687 pp. 25-36.

[20] A. S. Rao, "AgentSpeak(L): BDI agents speak out in a logical computable language", In: Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world (MAAMAW '96), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996, pp. 42-55.

[21] Agent Factory Agent Programming Language. URL: http://www.agentfactory.com/index.php/Main_Page, Accessed June 15th 2011

[22] FIPA, ”FIPA SL Content Language Specification”, 2002. URL: http://www.fipa.org/specs/fipa00008/SC00008I.html, Accessed June 15th 2011

[23] O. Khriyenko, "4I (FOR EYE) Technology: Intelligent Interface for Integrated Information", In: Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS-2007), Funchal, Madeira – Portugal, 12-16 June 2007, pp. 278-281

[24] O. Khriyenko, "Context-sensitive Multidimensional Resource Visualization", In: Proceedings of the 7th IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2007), Palma de Mallorca, Spain, 29-31 August 2007, pp 147-153

[25] O. Khriyenko and V. Terziyan, "A Framework for Context-Sensitive Metadata Description", In: International Journal of Metadata, Semantics and Ontologies, Inderscience Publishers, ISSN 1744-2621, 2006, Vol. 1, No. 2, pp. 154-164.

[26] L. Sauermann and R. Cyganiak, “Cool uris for the semantic web - w3c interest group note”. http://www.w3.org/TR/cooluris/, 2008, Accessed June 15th 2011

[27] O. Khriyenko, "Context-sensitive Visual Resource Browser", In: Proceedings of the IADIS International Conference on Computer Graphics and Visualization (CGV-2008), Amsterdam, The Netherlands, 24-26 July 2008, pp. 227-232

[28] O. Khriyenko, "4I (FOR EYE) Multimedia: Intelligent semantically enhanced and context-aware multimedia browsing", In: Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP-2007), Barcelona, Spain, 28-31 July 2007, pp. 233-240

[29] A. Naumenko, “SEMANTICS-BASED ACCESS CONTROL: Ontologies and Feasibility Study of Policy Enforcement Function”, In: J. Filipe and J. Minguillon (Eds.), Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST-07), March 3-6, 2007, Barcelona, Spain, pp. 150–155

[30] O. Khriyenko, S. Nikitin, and V. Terziyan, “Context-Policy-Configuration: Paradigm of Intelligent Autonomous System Creation”, In: Joaquim Filipe and Jose Cordeiro (Eds.), Proceedings of the *12th International Conference on Enterprise Information Systems (ICEIS-2010)*, 8-12 June, 2010, Funchal, Madeira - Portugal, ISBN: 978-989-8425-05-8, pp. 198-205.

