From Linked Data and Business Intelligence to Executable Reality

Vagan Terziyan
Department of Mathematical Information Technology,
University of Jyvaskyla
P.O. Box 35 (Agora), 40014, Jyvaskyla, Finland
e-mail: vagan@jyu.fi

Olena Kaykova
Industrial Ontologies Group, Agora Center,
University of Jyvaskyla
P.O. Box 35 (Agora), 40014, Jyvaskyla, Finland
e-mail: olena@cc.jyu.fi

Abstract—**This paper presents the concept of "Executable Knowledge", which is based on Linked Data and in addition to traditional subject-predicate-object semantic triplet model it contains also subject-predicate-query triplets. Actual values for such "executable" properties are supposed to be queried or/and computed whenever requested "on-the-fly" from/by some internal or external information source or computational capability provider at the right time and place according to the dynamic user context. We discussed two possible applications of this concept. One, which we named "Executable Reality", will enhance emergent (Mobile) Augmented and Mixed Reality concepts within two dimensions: utilization of Linked Data and Business Intelligence on top of it. Executable Reality will provide a real-time context-aware analytics related to various real-life objects selected by the users from their terminals. Other executable knowledge benefits are shown in the context of educational quality assurance and related to personalized online quality evaluation and ranking of various academic resources (people, departments, universities, etc.). It is demonstrated that the special Quality Assurance Portal for higher education may automatically utilize business analytics on top of Linked Data in the form of executable knowledge.**

Keywords- Business Intelligence; Linked Data; Mixed Reality; Executable Reality; Executable Knowledge; Quality Assurance

I. INTRODUCTION

Business intelligence (BI) can be considered as a set of methods, techniques and tools utilized on top of business data to compute (acquire, discover) additional (implicit) analytics out of it and to present it in a form suitable for decision-making, diagnostics and predictions related to business. Taking into account that "business data" is becoming highly heterogeneous, globally distributed (not only in the Internet space but also in time), huge and complex, extremely context sensitive and sometimes subjective, the ways the BI is utilized have to be qualitatively changed. Semantic (Web) Technology [1][2][3][4] is known to be a suitable approach to enable more automation within BI-related data processing. The vision of BI 2.0 [5] includes also issues related to Service-Oriented Architecture (SOA), mobile access, context handling, social media, etc. All these issues will also definitely benefit from adding semantics [6][7]. It is however a known fact that there is not much semantically annotated data available for BI. We have to live with data sets created independently, according to different schemas or even data model types. The realistic role of Semantic Technology for such data would be linking related "pieces" of it with some semantic connections and by doing this transforming the original data into the Linked Data.

There are no doubts that such semantically interlinked "islands" of data have a lot of hidden (implicit) and potentially interesting information that none of the separate data sets has alone. Now the challenge would be to utilize BI on top of Linked Data to be able to get all the benefits from semantic enhancement of the data.

Another trend is related to fast development of technology for better delivery and visualization of information. Among those there are Augmented Reality [12] and Mixed Reality [13] technologies and their mobile versions [14][15][16]. They are based on automated interlinking of various Web-based digital data collections with the real-time data from sensors about physical world and presenting it in a useful form for a user. An interesting topic would be considering these technologies in the context of business data or even BI-provided analytics. This may inspire more professional use of Augmented and Mixed Reality in addition to public use of it.

In this paper we propose "Executable Reality" as an enhancement of the "Mixed Reality" concept within two dimensions (utilization of Linked Data and BI on top of it). We present "Executable Knowledge" as a tool to enable Linked Reality and "Executable Focus" to control it by a user. Executable knowledge in addition to subject-predicate-object semantic triplet model (in ontological terms) contains also subject-predicate-query triplets ("executable properties"). Actual value for the properties based on a new triplet will be computed "on-the-fly" (based on user request navigated by executable focus) by some online BI service or other computational capability provider at the right time and place and according to the dynamic user context.

The rest of the paper is organized as follows: in Section II we discuss Linked Data issues and its enhancement by context-sensitive similarity links; in Section III we present (Mobile) Augmented and Mixed Reality technology and challenges; In Section IV we show how these technologies can be further developed towards "Executable Reality" on top of enhanced Linked Data and BI services (there we also present the concept of "Executable Knowledge"); we discuss Related Work in Section V; show one implementation of Executable Knowledge related to educational quality assurance in Section VI, and we conclude in Section VII.

II. LINKED DATA, CO-REFERENCE AND SEMANTIC SIMILARITY

Linked Data is a concept closely related to the Semantic Web yet providing some specific facet to it. According to Tim Berners-Lee “The Semantic Web is not just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data” (http://www.w3.org/DesignIssues/LinkedData.html). The so called “5 stars” advice from Tim Berners-Lee to enable Linked Data includes: making data available on the web (whatever non-proprietary format) as machine-readable structured data, utilizing open standards from W3C (RDF and SPARQL) to identify things and finally linking the data to other people’s data to provide context.

According to Kingsley Idehen (OpenLink Software CEO), due to development of Semantic Technology, meshing (or natural data linking) will replace mashing (brute-force data linking) and therefore mesh-ups can be considered as a next step comparably to the mash-ups in the sense to merge and integrate different data sources and processing devices to provide new information services.

Linked Data can be considered as an outcome of the technology, which semantically interconnects heterogeneous data “islands” (e.g., as shown in Figure 1). Even if the original sources of data are highly heterogeneous (not just only different schema of data within the same data model type but also different data model types), still it is possible to build some “bridges” between entities from these data sources utilizing semantic technology. The traditional Semantic Web approach would be: (a) creating a semantic model of the domain (ontology), (b) replacing original data from each source with full semantic (RDF) representation of its resources in terms of the ontology. Of course such an approach enables seamless integration of the original data and simplifies the usage of it. However with distributed and dynamic sources of data, which are managed and constantly updated independently, it would be difficult to provide such “semantic synchronization” (updating metadata and mapping it to the ontology) in real time. Therefore Linked Data would be less ambitious and the more practical approach would be: data sources are managed independently as they used to be; semantic connections between appropriate resources from different sources will be either automatically discovered or manually created whenever appropriate. Usage experience and usability performance for each separate data source will be preserved. The usability of such “virtually integrated” data storages will increase with the increase of the amount of the semantic “bridges”.

According to [8] there are three important types of RDF links within Linked Data:

(a) “relationship links” that point at related things in other data sources (like “object properties” in terms of OWL: *owl: ObjectProperty*);

(b) “identity links” that point at URI aliases used by other data sources to identify the same real-world object or abstract concept (e.g., *owl:sameAs*, *owl:sameIndividualAs*, *owl:equivalentClass*);

(c) vocabulary links that point from data to the definitions of the vocabulary terms that are used to represent the data (like “datatype properties” in terms of OWL: *owl:DatatypeProperty*).

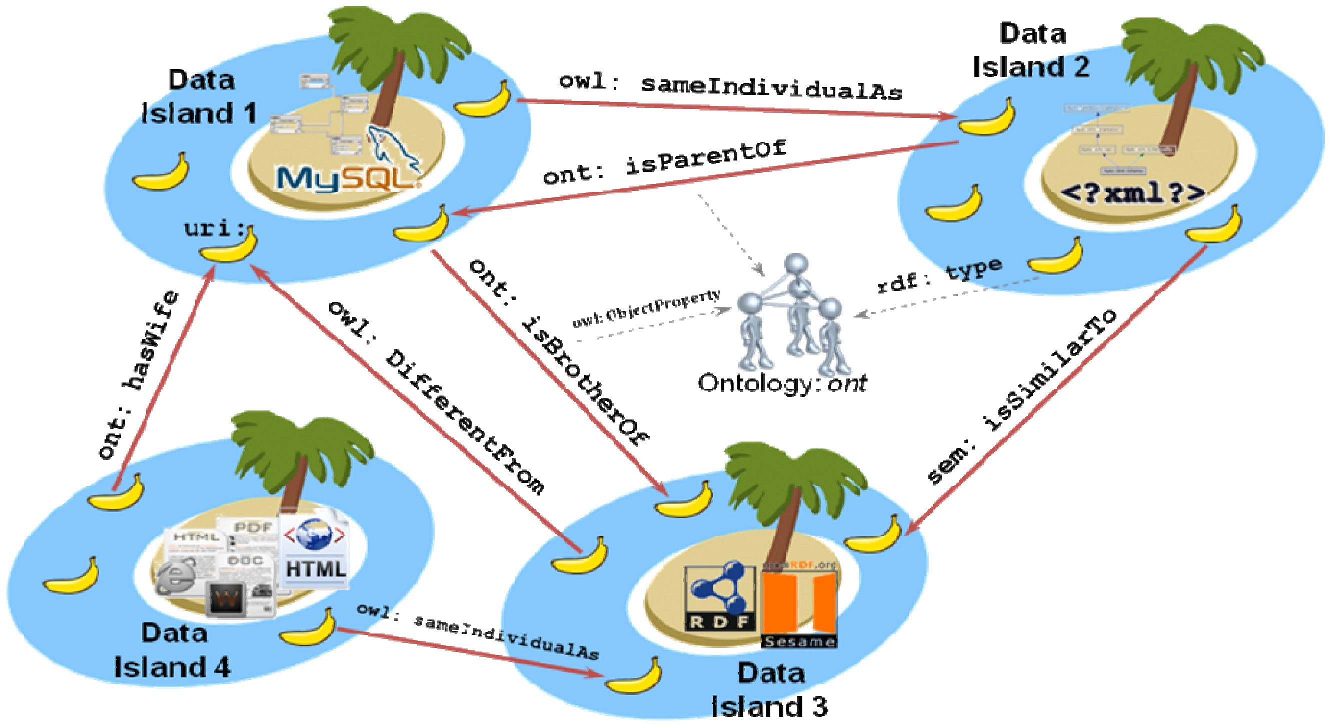


Figure 1. Linked Data: “bridges” between heterogeneous “islands” of data

We think it would be reasonable to extend traditional explicit semantic links within Linked Data with the implicit ones, e.g., those, which could be automatically derived by various reasoners. Among those special attention should be paid to the "semantic similarity" links. Usually, when someone queries a data, she looks for the resource(s), which are "the same" as the one specified in the query. However often there are none of such found. In many cases there is a sense to find "similar" resources to the target one. Similarity search was always a big issue within many disciplines and it is especially important for the Linked Data. The reason is related to the fact that actually same resources in different "islands" of data may have different URIs and often quite extensive work should be done to recognize same resources. Usually first we see that some resources look similar and therefore in practice could be the same ones and then we perform some check on the identity of the resources.

Semantic similarity search is not a trivial task because it should take into account heterogeneity of data types representing properties of the resources being compared, i.e., it should have a special distance calculator for each particular data type, then some normalization function for component distances (for each attribute of the compared resources) and finally some aggregation function of the component distances into the final distance. Also, depending on the context, different attributes of the resources may have different importance (weight) for the final distance aggregation function. Sometimes the context may influence not only the importance of an attribute but even the choice of the distance function itself.

Different sources of information (even isolated ones) are using the same words (concepts) from the real world to refer to particular groups of objects, people, events, etc. To enable automated information processing and interoperability, the providers are using URIs to distinguish between different instances of the same concept and trying to guarantee that once defined URI for something will stay the same throughout the whole set of documents coming from the same data source across all its history. It is obvious however that multiple sources of Linked Data cannot afford mutual awareness and sharing of the URIs, which results in URIs ambiguity in a global scale. In the context of Linked Data, the problem of determination of equivalent URIs referring to the same concept or entity is commonly known as "co-reference resolution" [26]. The problem is not that simple and traditional approaches to connect appropriate instances of data with *owl:sameAs* relation are not always working as shown in [27]. Also in [28] authors argue that in some contexts the comprehensive inference based on *owl:sameAs* relation for co-referenced entities is not possible due to hidden variations of the *owl:sameAs* semantics.

One of the well-known areas, in which co-reference becomes a major problem, is in author disambiguation [29]. There are many authors who share the same name and distinguishing between them is a vital part of any digital library or citation system. At the same time not only authors share the same names but variation in the spelling of names also leads to a single author having multiple identities (see example in Figure 2). This example is related to the academic career history of some female researcher. First stage of her career happens to be in USSR (Russian-speaking environment) and therefore first records on her identity (name, affiliation, etc.) and academic record (degrees, publications, projects, etc.) appear in the Web in Russian (Cyrillic letters used as it can be seen from downside of the Figure). Later she got an international passport where her name was transliterated from Cyrillic "Кайкова" into Latin "Kaikova". Her publication record since then has been indexed by Google Scholar according to this new identity and therefore Кайкова and Kaikova start to exist as two different persons (different URIs). Later, when time comes to change old international passports to the new ones, the transliteration rules changed and the new version of the name appear to be "Kaykova". After that all new publications of the researcher has been indexed by Google Scholar based on this new identity, which means that for this Web service there exist 3 different persons, which in reality is the same one (Figure 2). Assume that some Web application (e.g., the one making BI-driven academic quality summary report on some university) is going to automatically check citation index (e.g., h-factor) of this person and makes automatic query to Google Scholar with her current identity (Kaykova). It will get a number (e.g., h=10) based on incomplete publication record. Even if to manually make all 3 queries for "Кайкова", "Kaikova" and "Kaykova" and get 3 outcomes, e.g., h1=5, h2=18, h3=10, there is no way to automatically compute overall h-factor without analyzing content of all 3 publication sets. The problem actually is more complicated because all the 3 names may also belong to some other persons.

Summarizing the co-reference problem: 1) The same resource (e.g., a person, which has some record published in the Web) may have different URIs in different Web documents or databases; 2) Different resources may happen to be represented in some Web records with the same URIs due to similar identities; 3) BI-based computing reports being made separately on top of the records belonging to the same resource may not be integrated afterwards easily (if at all); 4) Explicit co-reference knowledge on similarity among resources, e.g., *owl:sameIndividualAs* network among distributed URIs, would be helpful; 5) Automated discovery of same resources in distributed records is not a trivial task; 6) Creating globally shared repositories of all Web resources with their identities is not a trivial task either (if realistic at all); 7) Relations like *owl:sameIndividualAs* may have different hidden semantics in different contexts (time, location, goal, preferences, etc.) and therefore should be carefully analyzed against the context when applied; 8) The co-reference problem handling is the one of the great importance for future potential of Business Intelligence, which is expected to be automatically applied for the Linked Data utilization.

Figure 2. Co-reference problem visualized

OKKAM (www.okkam.org), as a Large Scale Integrating Project (January 2008 - June 2010) co-funded by the European Commission, was looking for a scalable and sustainable solution for systematic and global identifier reuse in decentralized information environments enabling users to get and create globally shared URIs [30]. Created so far OKKAM repository of about 7.5 million entities cannot however solve co-reference problem at a full scale to be used by Linked Open Data community.

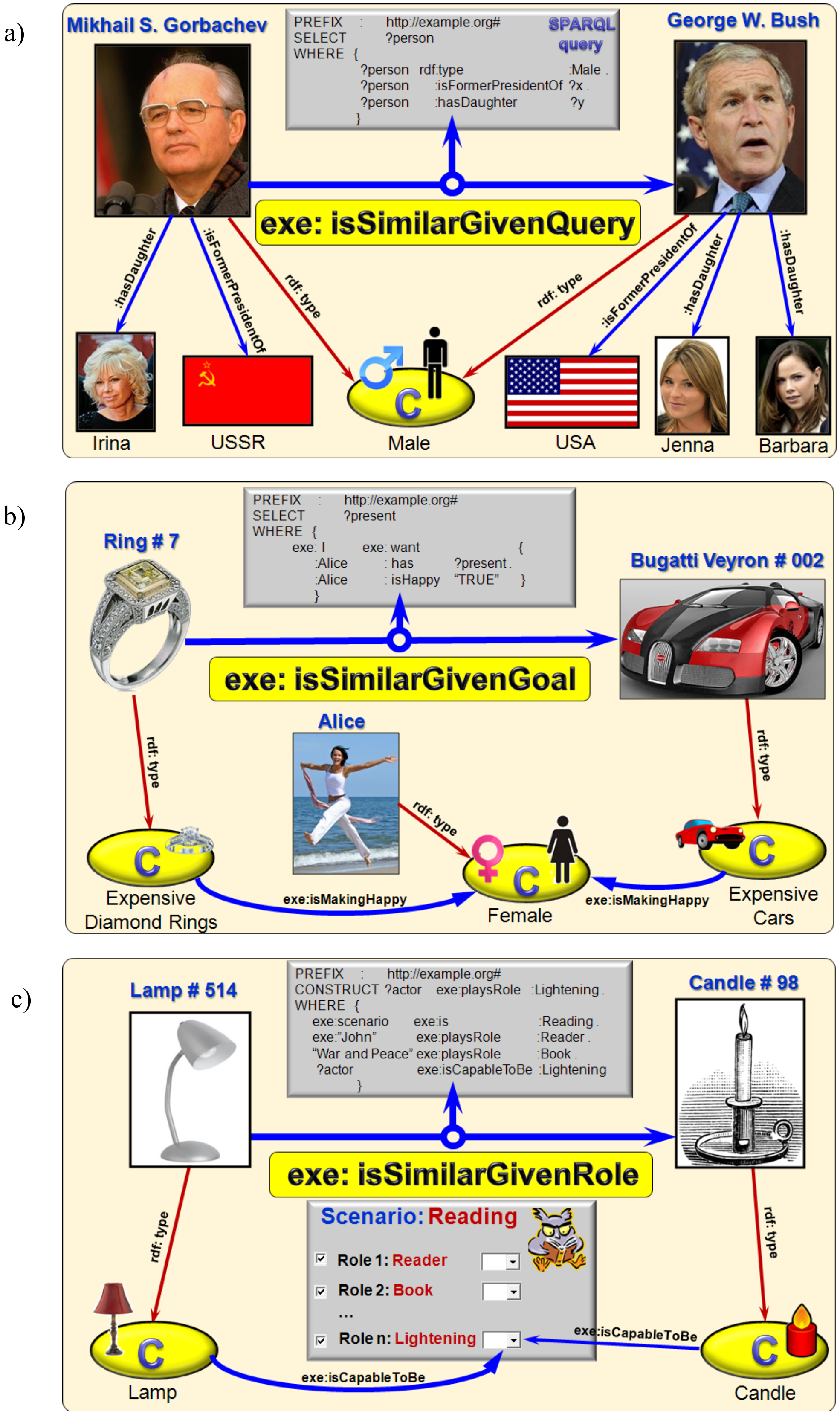
Other use of similarity measure (than co-reference resolution) is when one is looking for a capability-providing resource (e.g., a service), cannot find exactly the one she wants but still will be satisfied by finding a resource with similar functionality. Therefore it would be reasonable to have some explicit similarity links between stored data entities obtained as a result of appropriate similarity search procedures. The two major challenges here are: (a) a resource within one data "island" may have very different model of description when compared to some resource within another data "island" (e.g., a human documented in a relational database will not be easily compared with a human from some XML storage or from some html document); (b) some resources being very different in one particular context could be considered as similar ones in some other context.

We consider three types (sub-properties in terms of OWL) of semantic similarity based on common ternary object property relation named *isSimilarGivenContext* and they are: (1) *isSimilarGivenQuery*; (2) *isSimilarGivenGoal*; and (3) *isSimilarGivenRole.*

The first type of similarity assumes that two resources can be considered as similar ones (in the context of some semantic query, e.g., SPARQL query) if this query, being applied over the locations of these two resources, returns both of them as a result. See example in Figure 3(a). Here the resources "Mikhail S. Gorbachev" and "George W. Bush" are shown to be inferred as the similar ones, given query "Former president, male with at least one daughter".

The second type of similarity applies to the resources which can be replaced with each other as input parameters needed to perform some function (action) or utilize some external capability (product or service) for achieving some goal without affecting expected outcome. For example, a "Bugatti Veyron" car would be a similar resource to e.g., "Expensive diamond ring" as an "input" ("making wedding present"), given goal "To make the girlfriend happy" as it is shown in Figure 3(b).

The third type of similarity assumes that two resources will be used as similar ones if they both can fill some slot in a business process with the specified role. Consider the example in Figure 3(c). Here some resource (instance of class "Lamp") has been computed as similar one to another one (instance of class "Candle"), given role "Lightening".



```
PREFIX   :    http://example.org#
SELECT        ?person
WHERE  {
          ?person   rdf:type                :Male .
          ?person      :isFormerPresidentOf ?x .
          ?person      :hasDaughter           ?y
       }
```

```
PREFIX   :    http://example.org#
SELECT        ?present
WHERE  {
       exe: I        exe: want             {
           :Alice        : has        ?present .
           :Alice        : isHappy    "TRUE"     }
       }
```

```
PREFIX     :     http://example.org#
CONSTRUCT  ?actor    exe:playsRole  :Lightening .
WHERE  {
   exe:scenario       exe:is                :Reading .
   exe:"John"         exe:playsRole         :Reader .
   "War and Peace"    exe:playsRole         :Book .
    ?actor            exe:isCapableToBe     :Lightening
         }
```

Figure 3. Three types of similarity relations: a) similar in the same context; b) similar to be used to reach the same goal; c) similar when playing the same role

More information about our approach for defining context in various practical applications and semantic similarity search within context can be found in [9][10][11].

The major challenge is how to provide support for automated utilization of Linked Data, which in fact remains heterogeneous, and how to get added value of additional semantic connections between data components. Anyway we claim that providing similarity links, in addition to traditional types of RDF links described in [8], can be very helpful for practical utilization of Linked Data and we will try to show this in the following sections of the paper.

III. AUGMENTED AND MIXED REALITY

Augmented Reality (AR) [12] is a technology aimed to enhance the traditional perception of a reality (real-world environment), which elements are augmented by computer-generated sensory input (e.g., data, sound or graphics). AR enriches real world for the user rather than replaces it. By contrast, *virtual reality* replaces the real-world with a simulated one. Emerging development of mobile computing has naturally resulted to growing interest towards Mobile AR [15] and also to Ubiquitous Mobile AR [14] for successfully bridging the physical world and the digital domain for mobile users. The AR concept has been further developed to Mixed Reality [13], which means merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real time. In June 2009, Nokia Research Center announced the vision [16] of Mobile Mixed Reality, according to which a phone becomes a "magic lens" (smart and context-aware), which lets users look through the mobile display at a world that has been supplemented with information about the objects that it sees. The users simply point their phone's camera, and look "through" the display. Objects of interest visible in the current view will be gathered from existing Point-Of-Interest databases or created by the user and will be highlighted. They can be associated with physical objects or featureless spaces like squares and parks. Once selected, objects provide access to additional information from the Internet and hyperlinks to other related objects, data, applications and services. Context-awareness is guaranteed by various rich sensors that are being incorporated into new phones (GPS location, wireless sensitivity, compass direction, accelerometer movement, sound and image recognition, etc.). Therefore the new technology is going to actively utilize acquired dynamic context to better filter and select relevant information about surrounding real-world objects for a user.

In the following section we further develop the concept of (mobile) mixed reality within two dimensions: the first one is related to Linked Data utilization and the second one will be related with the utilization of Business Intelligence through "Executable Knowledge".

IV. TOWARDS "EXECUTABLE" REALITY

The concept of "Executable Reality" and associated technology, which we are offering, should be considered as an extension of the (Mobile) Mixed Reality concept and the technology. If the traditional technology assumes that the explicitly available relevant data about some real-world object will be taken from some database and delivered to the user on demand (based on her attention focus pointed to this object), the Executable Reality in addition is able to provide online BI computation based on similar demand (we call it "Executable Focus") and present to the user results of computed analytics adapted for the current context.

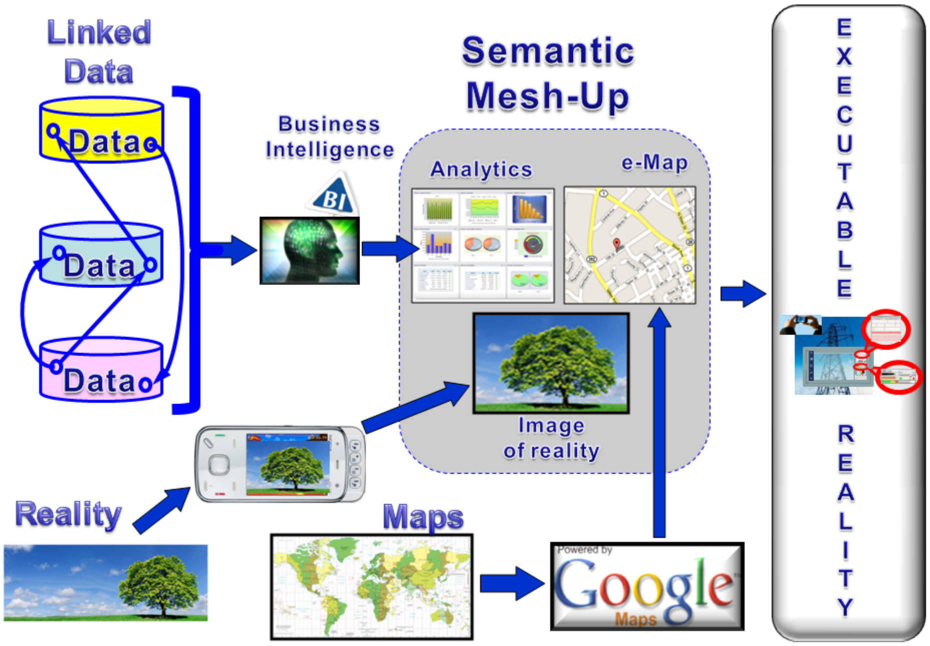
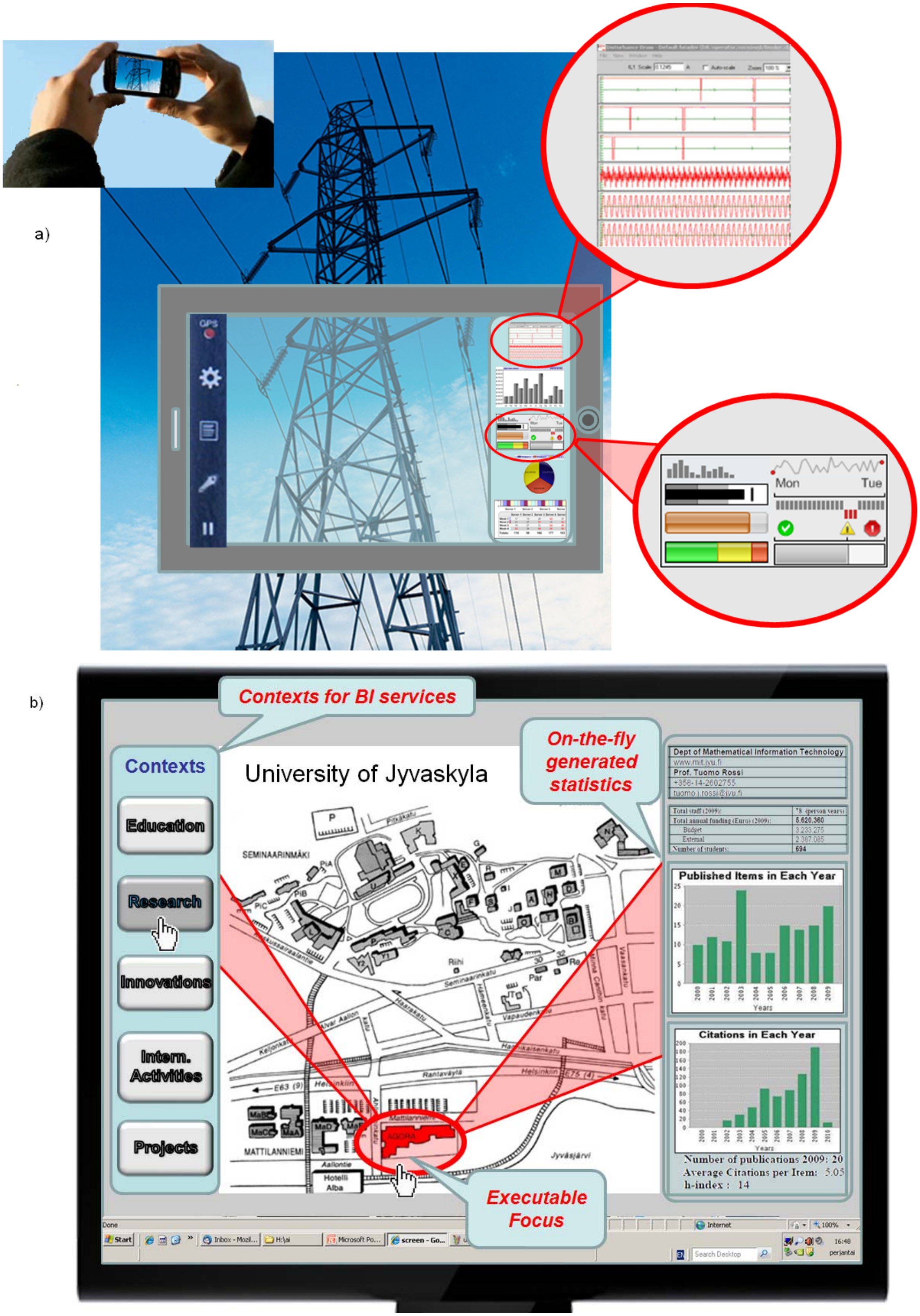


Figure 4. Executable-Reality-related process illustrated



Dept of Mathematical Information Technology
www.mit.jyu.fi
Prof. Tuomo Rossi
+358-14-2602755
tuomo.j.rossi@jyu.fi

Total staff (2009):	78 (person years)
Total annual funding (Euro) (2009):	5.620.360
Budget	3.233.275
External	2.387.085
Number of students:	694

Number of publications 2009: 20
Average Citations per Item: 5.05
h-index : 14

Figure 5. Executable Reality use case examples: (a) on-the-fly computed statistics about power line performance is delivered to mobile terminal of the maintenance engineer on implicit demand; (b) research performance statistics is delivered to the user based on chosen unit (click on the building where the university department is located and selecting context “research” for filtering appropriate data from the unit needed for research performance calculations)

Consider an architecture of a typical Executable Reality process in Figure 4. The content of the three information channels has been mixed in a kind of "mesh-up" (assuming that similar resources from all the channels are recognized, identified and semantically mapped together). The first channel presents the "image of reality" taken through the focus of some sensor (e.g., mobile phone camera). The second channel presents the geographic information (e.g., received online via Google Maps service). Linking the content of these two channels together provides a typical case of the Mobile Mixed Reality. Let us however consider the third channel as shown in Figure 4. The initial sources of its content are assumed to be heterogeneous, distributed and interconnected according to the Linked Data layer of semantics. In contrast to the Mixed Reality, the Executable Reality processes are not only capable to map the actual content from the Linked Data original sources into the structured image of Reality, but they are also capable to map the analytics (outcomes of BI software-as-a-services) taken automatically from the Linked Data into the image of Reality. A typical interface from an Executable Reality application will look as follows. Some sensor (e.g., mobile phone camera) gives to a user a snapshot of the Reality. The user clarifies a focus of her attention (e.g., selects certain object highlighted on the snapshot), "clicks" on it (directly or through special additional controls) and as a result, instead of getting just traditional (e.g., Mobile Mixed Reality) information output, she will get also some analytics associated with the selected object based on information queried and processed on-the-fly from the Linked Data.

Two use-case scenarios for the Executable Reality are shown in Figure 5 (a, b). In the first one, the user (maintenance engineer of the power network company) is putting the executable focus (smart phone camera) into the direction of the power line and by doing this makes implicit request (associated with the profession of the user, knowledge about the context and the type of resource recognized) for, e.g., the last 24 hours performance statistics of this power line. The query will go further to the server; appropriate BI service will be selected (based on semantic comparison of the query and available service descriptions) and automatically invoked; a resulting page with numbers, graphics (and sounds if appropriate) will be generated and delivered back to the terminal and shown in the appropriate window of the screen as shown in Figure 5 (a).

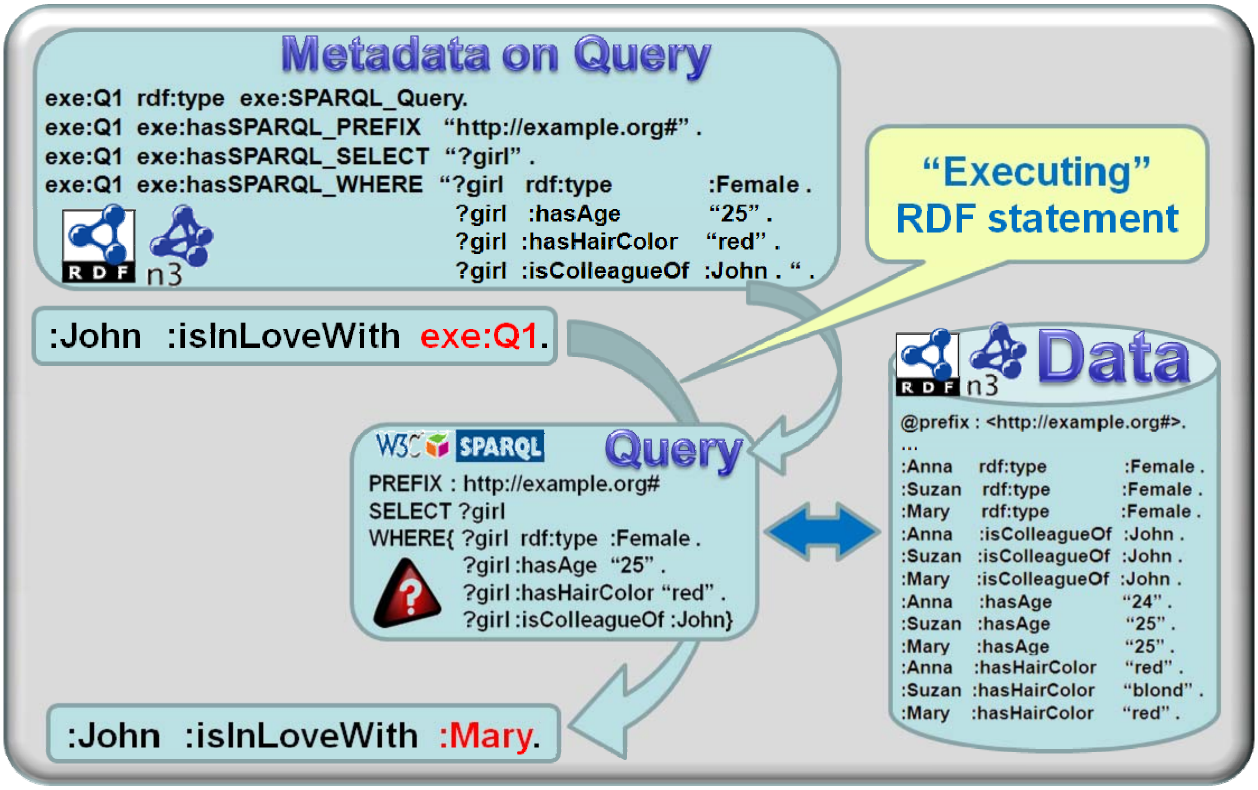
Another scenario in Figure 5 (b) shows that a user observes the campus map of some university and selects the building where a particular department is located. The user also selects the context in which she wants to get performance statistics of the unit, e.g., "research". Chosen object and the context together form the executable focus, which will automatically generate the query for the required computation. Then the process goes in a similar way as with previous scenario and the user will get "fresh" statistics (assuming that some remote Web-services, i.e., some public citation indexes collectors will be automatically queried and processed) concerning performance of the department.

To enable such scenarios we have to find an effective way to utilize Linked Data, which is a natural source for online BI computations, and also to enable BI functionality as semantically annotated Web services. We propose to organize Linked Data in form of "Executable Knowledge".

Executable Knowledge can be considered as distributed (or organized as a cloud) set of heterogeneous data storages and computational services (e.g., BI) interconnected with semantic (RDF) links. The major feature here is that, in addition to the traditional ("subject-predicate-object" or "resource-property-value") triplet-based semantics of an RDF link (either "datatype" property, where "value" is a literal; or "object" property, where "value" is another resource), the new model will have new property type named "executable property" with the structure: "subject-predicate-query". It is supposed that reasoners, engines, etc., working with such knowledge will execute the query within the target triplet and treat the obtained result as a value for the property. Two immediate advantages of this extension are: (a) the triplet will always implicitly keep knowledge about most recent value for the property because the query to some data storage or to some BI function will be executed only on demand when needed and the latest information will be delivered; (b) the query may be written according to different standards, data representation types, models and schemas so that heterogeneity of original sources of data and capabilities will not be a problem. Therefore distributed data collections can be maintained independently (autonomously) and "queried" in real time by executable RDF links.

Consider an example in Figure 6. Here the executable statement in RDF (N3 syntax) means: "If you want to know with whom John is currently in love, execute the query Q1". The query Q1 (prefix "exe:" points to Executable Knowledge ontology and indicates that the RDF statement is executable) in this case is semantically described as a SPARQL query to the RDF data storage and it means: "Select the girl from the current database of staff, who is colleague of John, has red hair and is 25 years old". When the SPARQL query engine executes the query and finds that "Mary" fits it, the executable RDF statement is transformed into the traditional one (reference to the query "exe:Q1" is replaced with actual value "Mary"). Notice that, if the same knowledge will be explored after 1 year, then the same executable statement will be transformed into: "John is in love with Anna", because staff data (separate source) will be autonomously updated (Anna becomes 25 years old) and RDF connections (semantic layer of Linked Data) on top will be automatically updated when executed.

Consider similar example in Figure 7 and notice that here we have an SQL query to some relational database as implicit value of executable RDF statement. The query Q2 means request for computing average journal papers' publication performance of young (< 30) PhD students. The original executable RDF statement means: "If you want to know average performance of young doctoral students in AI Department, execute query Q2". When the query returns computed value, the executable RDF statement is transformed into the traditional one (reference to query "exe:Q2" is replaced with actual value "7", which means that the "executable" RDF property is replaced with the "datatype" property).



```
exe:Q1  rdf:type  exe:SPARQL_Query.
exe:Q1  exe:hasSPARQL_PREFIX   "http://example.org#" .
exe:Q1  exe:hasSPARQL_SELECT   "?girl" .
exe:Q1  exe:hasSPARQL_WHERE   "?girl  rdf:type          :Female .
                               ?girl  :hasAge           "25" .
                               ?girl  :hasHairColor     "red" .
                               ?girl  :isColleagueOf  :John . " .
```

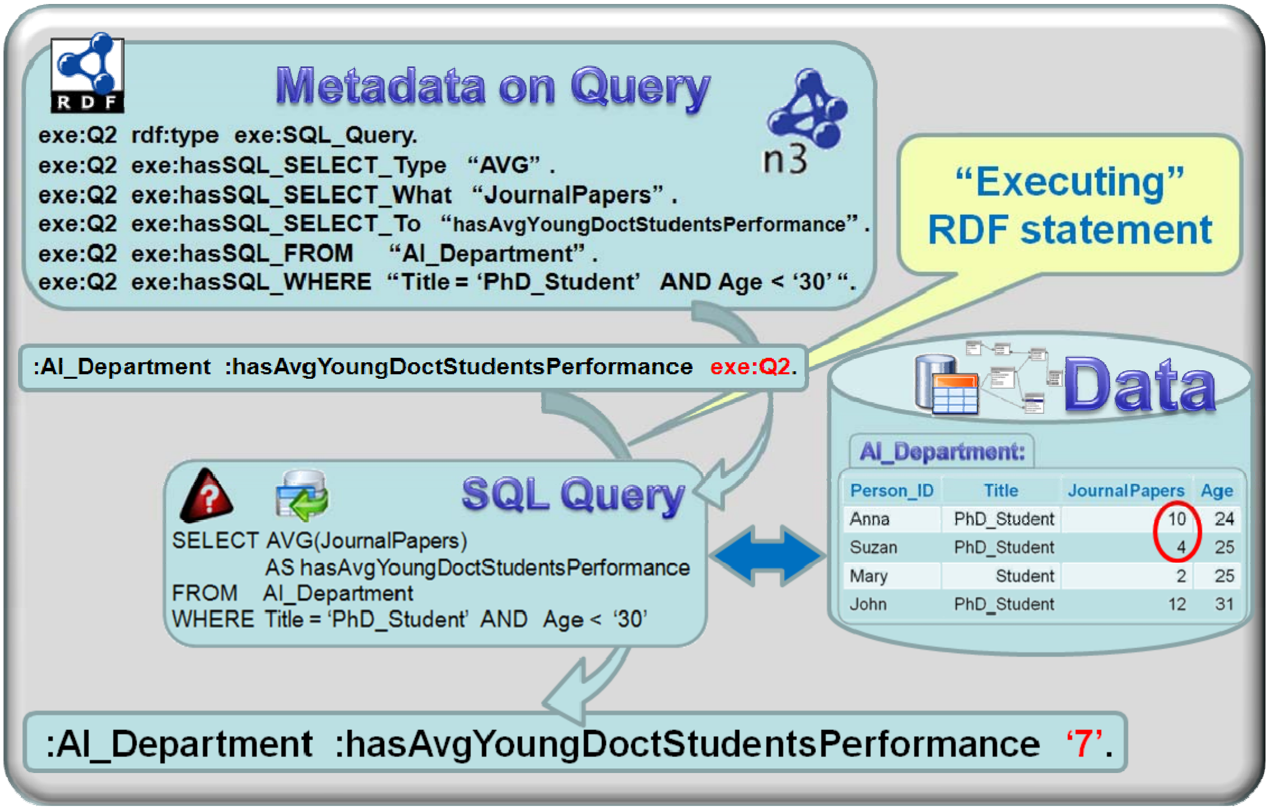
```
:John  :isInLoveWith  exe:Q1.
```

```
PREFIX : http://example.org#
SELECT ?girl
WHERE{ ?girl  rdf:type  :Female .
       ?girl :hasAge  "25" .
       ?girl :hasHairColor "red" .
       ?girl :isColleagueOf :John}
```

```
@prefix : <http://example.org#>.
...
:Anna    rdf:type          :Female .
:Suzan   rdf:type          :Female .
:Mary    rdf:type          :Female .
:Anna    :isColleagueOf  :John .
:Suzan   :isColleagueOf  :John .
:Mary    :isColleagueOf  :John .
:Anna    :hasAge           "24" .
:Suzan   :hasAge           "25" .
:Mary    :hasAge           "25" .
:Anna    :hasHairColor     "red" .
:Suzan   :hasHairColor     "blond" .
:Mary    :hasHairColor     "red" .
```

```
:John  :isInLoveWith  :Mary.
```

Figure 6. Example of processing executable RDF statement, which contains implicit value as SPARQL query to the RDF storage



```
exe:Q2  rdf:type  exe:SQL_Query.
exe:Q2  exe:hasSQL_SELECT_Type   "AVG" .
exe:Q2  exe:hasSQL_SELECT_What   "JournalPapers" .
exe:Q2  exe:hasSQL_SELECT_To   "hasAvgYoungDoctStudentsPerformance" .
exe:Q2  exe:hasSQL_FROM     "AI_Department" .
exe:Q2  exe:hasSQL_WHERE  "Title = 'PhD_Student'   AND Age < '30' ".
```

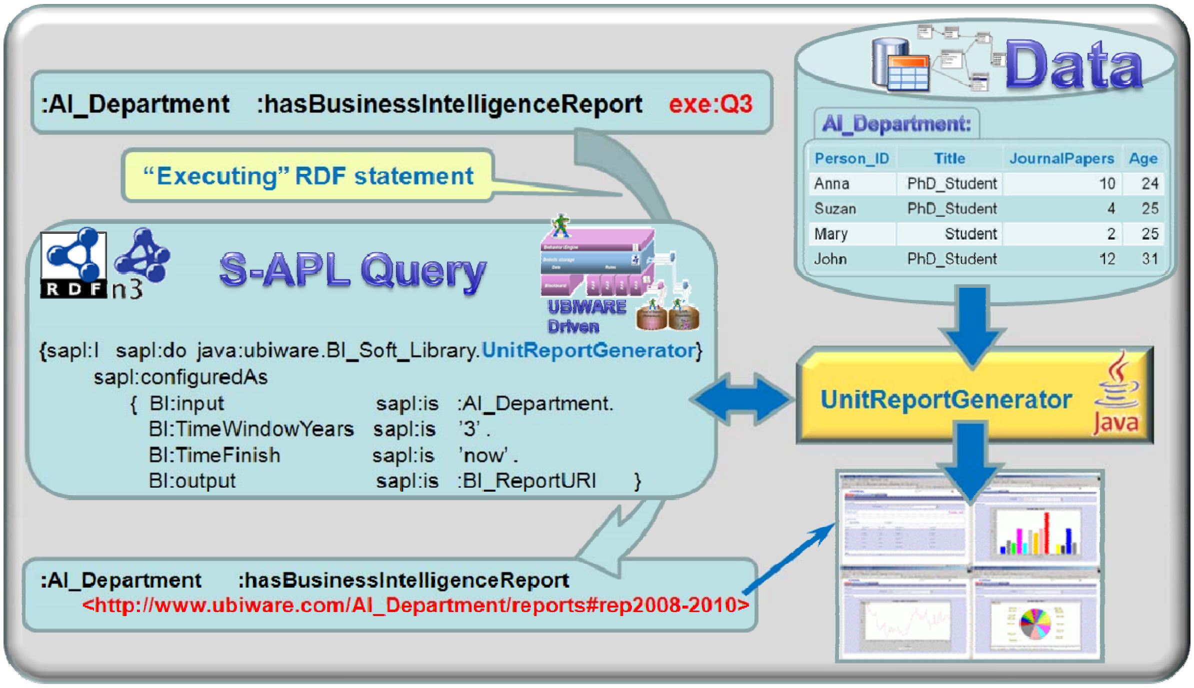
```
:AI_Department  :hasAvgYoungDoctStudentsPerformance  exe:Q2.
```

```
SELECT AVG(JournalPapers)
          AS hasAvgYoungDoctStudentsPerformance
FROM    AI_Department
WHERE  Title = 'PhD_Student'  AND   Age <  '30'
```

Person_ID	Title	JournalPapers	Age
Anna	PhD_Student	10	24
Suzan	PhD_Student	4	25
Mary	Student	2	25
John	PhD_Student	12	31

```
:AI_Department  :hasAvgYoungDoctStudentsPerformance  '7'.
```

Figure 7. Example of processing executable RDF statement, which contains implicit value as SQL query to a relational database



```
:AI_Department   :hasBusinessIntelligenceReport   exe:Q3
```

AI_Department:

Person_ID	Title	JournalPapers	Age
Anna	PhD_Student	10	24
Suzan	PhD_Student	4	25
Mary	Student	2	25
John	PhD_Student	12	31

```
{sapl:I   sapl:do  java:ubiware.BI_Soft_Library.UnitReportGenerator}
      sapl:configuredAs
          {  BI:input                  sapl:is   :AI_Department.
             BI:TimeWindowYears   sapl:is   '3' .
             BI:TimeFinish            sapl:is   'now' .
             BI:output                 sapl:is   :BI_ReportURI      }
```

```
:AI_Department     :hasBusinessIntelligenceReport
      <http://www.ubiware.com/AI_Department/reports#rep2008-2010>
```

Figure 8. Example of processing executable RDF statement, which contains implicit value as S-APL query to BI software as a service

Consider the example in Figure 8. Here we have an executable RDF statement that can be interpreted like: "If you want to get basic BI-statistics report for the AI Department for the last 3 years, execute query Q3". Behind this query there is a Java software module "UnitReportGenerator" provided as-a-service from online software library. The query itself is written in S-APL (Semantic Agent Programming Language [17]) used for UBIWARE-based applications [18]. S-APL is a RDF-based language for multi agent systems, in which both data and actions are described semantically. UBIWARE [19] ("Smart Semantic Middleware for Ubiquitous Computing") has been developed by Industrial Ontologies Group (http://www.cs.jyu.fi/ai/OntoGroup). It is a software technology and a tool to support design and installation, autonomic operation and interoperability among complex, heterogeneous, dynamic and self-configurable distributed systems, and to provide a coordination, collaboration, interoperability, data and process integration service. UBIWARE platform is used actually to deal with "Executable Knowledge" and its utilization for "Executable Reality" services. In the example, when the BI software is executed, it generates the html page where all analytics are visually presented with different BI widgets. The URI for such page will replace the implicit "exe:Q3" value from the original RDF statement and creates traditional RDF statement with object property connecting two resources (AI Department URI and BI statistics Report URI).

There is also a possibility to compute semantic similarity between resources from different data storages and automatically create appropriate RDF connections for similar (same) instances. As it was shown in Section II, some instances can be considered as similar ones in one context and can be considered otherwise in another context. Therefore, similarly to the examples above, the "Executable Knowledge" supports also RDF statements with implicit similarity search queries, in which needed query parameters are automatically taken from the current context. Change of context can be considered as implicit query (if appropriate setup is made) to re-compute similarity links, which makes the RDF graph on top of Linked Data very dynamic. Our approach for context-sensitive semantic similarity computing and its implementation is discussed in [11].

Mixed Reality is just one possible way to utilize Executable Knowledge concept. There should be definitely other application areas for it. Generally many industrial applications, which require dynamic self-configurable solutions, applications and architectures, will benefit from the flexible Executable Knowledge, as our experience with UBIWARE industrial cases demonstrates [19].

V. RELATED WORK

Concepts of "virtual", "augmented", "mixed", etc., realities discussed in Section III are being actively developed into various services for the public. There are many other relevant concepts and activities, which have many common features with the above, having however some specifics. One such abstraction is so-called "Mirror World" [20], which is a representation of the real world in digital form mapped in a geographically accurate way. Mirror worlds can be seen as an autonomous manifestation of digitalized reality including virtual elements. Another relevant concept is "Metaverse" (http://metaverseroadmap.org), which is the convergence of virtually-enhanced physical reality and physically persistent virtual space being a fusion of both. The “Second Life” (http://secondlife.com/) is a 3D virtual world enhanced by social networks and communication capabilities. “Lifelogging” [21] is continuous capturing from a human and sharing through the Web various data, events and activities collected by various devices, sensors, cameras, etc. Other slightly different concept is “Lifeblog” [http://europe.nokia.com/support/product-support/nokia-photos], which is also known as a popular service for collecting and putting into a timeline (mobile) user activities and creating data in the form of complex multimedia diary.

Our intention was to find out reasonable services out of these concepts suitable not just for public use but mostly for professionals. We explored the possibility to utilize BI as an additional capability for that purpose. Preliminary information on the interesting relevant effort named “Augmented BI” has appeared in the Web [22] quite recently. Augmented BI is considered in [22] as a process of using a mobile device to scan an image or a barcode and overlaying metrics and or charts onto the image. This supposes to facilitate the process of a store manager moving around a retail store, who would like to get more information about certain products’ sales performance. See Figure 9, which demonstrates a possible use case for the Augmented BI. There are some evident similarities with our use-cases from Figure 5, however our implementation benefits from the Linked Data utilization and allows context-sensitive view to the BI-enhanced reality.

Figure 9. Demonstration of possible Augmented BI usage scenario [22].

Our solution related to BI-enhanced mixed reality (or Executable Reality) is based on the Executable Knowledge concept. The Executable Knowledge inherits some features from a Dynamic Knowledge (see, e.g., [23]), which is actually dynamically changing knowledge and according to (www.imaginatik.com) providing on-demand, in-context, timely, and relevant information. Issues related to such knowledge include power and expressive tools and languages (such as, e.g., LUPS [24]) for representing such knowledge and proper handling of conflicting updates as addressed in [23]. Given an initial knowledge base (as a logic program) LUPS will sequentially update it.

Since executable knowledge is definitely a kind of dynamic knowledge, other issue would be whether it is declarative or procedural knowledge. A procedural knowledge (or knowledge on how to do something) is known to be a knowledge focused on obtaining a result and exercised in the accomplishment of a task, unlike declarative knowledge (propositional knowledge or knowledge about something) [25]. Procedural knowledge is usually represented as finite-state machine, computer program or a plan. It is often a tacit knowledge, which means that it is difficult to verbalize it and transfer to another person or an agent. The opposite of tacit knowledge is explicit knowledge.

The concept of executable knowledge can be actually considered as a kind of hybrid of declarative and procedural knowledge. For similar purpose, Marvin Minsky in [37] suggested to use so called “demons” within frame models already in 1975. Demons are supposed to be attached to some slots in a frame to cause execution of some procedure when accessed. Since that, however, demons have never been supported by the RDF data model. As it can be seen from the examples in Section IV, by “executing” knowledge, one actually transforms tacit (procedural) knowledge into explicit (declarative one). Therefore an executable knowledge contains explicit procedural (meta-) knowledge on *how to acquire* (or compute) declarative knowledge. Such capability means that the executable knowledge is naturally self-configurable knowledge (or more generally – self-managed knowledge). We use S-APL (Semantic Agent Programming Language [17]) for its representation, which is based on RDF (N3) syntax and which is equally suitable to manage declarative and procedural knowledge.

Our implementation of the executable knowledge on top of UBIWARE [18][19] agent-driven platform allows UBIWARE agents autonomously “execute” knowledge by following explicit procedural instructions for BI services execution and therefore updating (or making explicit) appropriate declarative beliefs.

The basic architecture of UBIWARE as a cloud-based platform is shown in Figure 10. It is supposed that any user of UBIWARE will be able to design and upload her own “application” via friendly and simple Web interface and this application can be executed and run continuously at the UBIWARE cloud. An application supposed to be designed in accordance with the SOA principles and it looks like semantic specification (in S-APL) of needed components (capabilities and knowledge as-a-service) and semantic specification (in S-APL) of desired business logic to connect these components. The components needed for the application can be internal (i.e., available in the cloud, semantically annotated, searchable and executable internally) and external (Web services, databases, etc.), for automatic utilization of which special semantic interfaces (adapters) are needed. The layers of knowledge between the application and the components in the Figure 10 are actually playing role of such adapters. These knowledge layers are organized as executable knowledge, which means that whenever a running application needs to refer to some internal or external component it simply sends semantic query to the executable knowledge and then the actual query (formal information or service request) will go from the executable knowledge to intended components discovered on-the-fly.

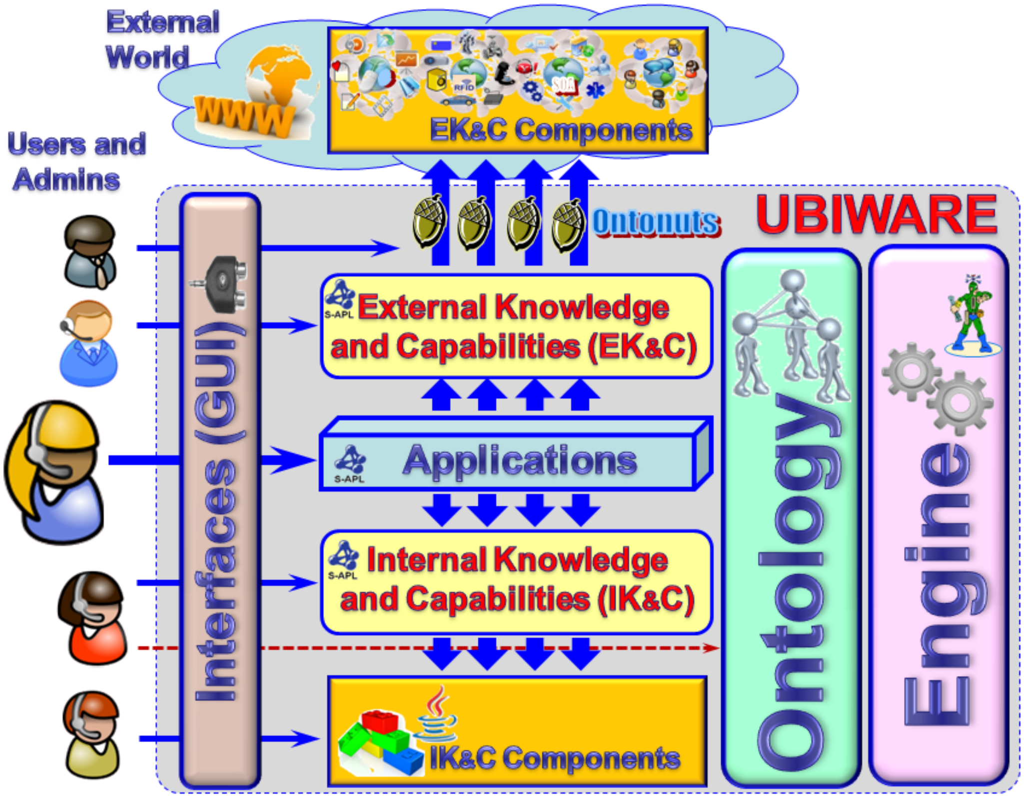


Figure 10. Architecture of the UBIWARE platform

The process of executing an application with discovered on-the-fly components is supported by special UBIWARE ontology and special agent-driven engine working as "Knowledge Processor" for knowledge execution.

Some external components, which are not software-as-a-service, may require special additional semantic interfaces to their APIs to be used automatically. In UBIWARE such interfaces (semantic software components) are named as *Ontonuts* and they are capable to facilitate the presentation of modular scripts and plans related to the external world utilization within the UBIWARE platform [36].

VI. EXECUTABLE KNOWLEDGE IN ACTION: QUALITY ASSURANCE CASE

One of the ongoing activities, which is actively utilizing the Executable Knowledge concept, is the EU Tempus-IV Project TRUST: "Towards Trust in Quality Assurance Systems" (516935-TEMPUS-1-2011) [31][32][33][34][35]. The overall goal of the TRUST project is to support the reforms of Ukrainian Higher Education (HE) by introducing a common, comprehensive and transparent Quality Assurance (QA) framework for all HE institutions (HEI) and QA organizations. The framework is based on the knowledge triangle ("education-research-innovation") and is open to stakeholders. An ecosystem of solutions is developed that enables, supports and automates the QA activities and transactions between HEIs, different national and international QA actors, students and different stakeholders and supports various forms of information exchange and knowledge sharing. The framework is assumed to guarantee trust between all QA players and society by ensuring that all QA procedures will be based on credible, transparent and relevant sources of information and explainable decision-making techniques documented in a common portal. Impact of each dimension of the knowledge triangle will be taken into consideration and the most independent and therefore credible sources from each dimension will be included into the system of quality criteria. Education quality is proposed to be assessed by both EU students who have taken courses or got degrees in Ukrainian HEIs and return back to EU and Ukrainian graduates moving to work or to continue study to EU. Research can be evaluated by official sources of international scientific citation indices mediated by Web-services. EU companies are to be involved into evaluation of innovation potential of Ukrainian HEIs. A trusted QA system should be based as much as possible on external objective evaluations. However because it is difficult to immediately utilize expensive experience of external evaluators in Ukrainian QA system one may (as the first step) make the academic data, metadata, quality indicators and QA processes available and transparent to national and international academic community and combine it with other publicly available information within an ecosystem based on a web portal which enables external assessment of the academic performance.

These objectives are supported by a flexible and powerful instrument – Portal [34][35][32], which is a work-in-progress, providing a set of solutions that (on the basis of Executable Knowledge) enable, support and automate the activities, information flows and transactions within the ecosystem of individuals, HEIs, and QA organizations. The core system is extended with mechanisms allowing consideration of flexible multidimensional and multicontextual quality indicators, which will reflect constantly variable contexts (caused by political, economical, etc., reasons) at the different user-dependent levels (international, European, national, local) in all aspects of HEIs processes. These enable each HEI or national QA organization to develop its own appropriate QA strategies, HEI evaluations, rankings, etc., and provide capabilities for self-proof of decisions. Provided IT-support of QA enables: machine-processable QA-related information; management of globally distributed and heterogeneous QA-related data collections and Web-services; QA-related automated knowledge transfer through intelligent information retrieval, extraction, sharing, reuse and integration. To achieve this, the knowledge needed for QA is organized according to the Executable Knowledge concept and it is augmented in several dimensions:

- To allow anybody to easily add her own QA technique (a "Quality Calculator") or evaluation criteria (i.e., executable properties as described in Section IV) to the knowledge base and to get a personalized view on the quality status (in absolute or relative scales) of any educational organization or any educational outcome. As a result such executable knowledge becomes in a way a "Smart Knowledge" (i.e., enable ranking, evaluation, etc., formulas, QA procedures and techniques to be proactive knowledge instances, to be self-descriptive, extendable, self-managed and reusable);
- To make the results more transparent and trustful such executable knowledge must also be a "Cross-Validated Knowledge" (i.e., providing Service-Oriented Architecture for the portal enabling automatic update of the values of various quality indicators by taking them from external Web-based sources (portals, databases, etc.), such as, e.g., ISI Web of Knowledge, Google Scholar, etc., externalizing and internationalizing various quality monitoring activities);
- To help the user see the reasons behind good or bad performance we need our knowledge to "behave" as a "Self-Explanatory Knowledge" (that provides automated support for detailed explanation of every calculated or inferred value of any quality indicator used in QA activities);
- To automate the interpretation of the values of various quality indicators in different situations we need such executable knowledge to perform also as a "Context-Aware Knowledge" (i.e., utilization of formalized knowledge about context (local, regional, national, international, etc., for providing more grounded evaluations in a particular context).

The multidimensional and multilayered formalized model of the executable knowledge (including smart, self-explanatory, cross-validated and context-aware knowledge) about QA domain (resources, parameters, values, activities, etc.) collected or linked during the TRUST project is called QA Ontology. It includes (a) core layer (the one, which specifies concepts and properties related to knowledge triangle: education, research and innovation and which supposed to be a required part for various quality evaluations); (b) customized layer (the one which every organization can flexibly adapt to a local context or every user can adapt to own preferences); (c) system of values (which defines weights for various quality indicators in various contexts); (d) QA processes (i.e., formally specified internal or cross-organizational processes to enable QA execution monitoring).

The essential components of the TRUST portal are presented in Figure 11. One can see that the information about educational resources (e.g., universities, departments, academic personnel, etc.) is automatically collected from remote but trusted sources of data and interlinked with the metadata layer based on co-reference resolution and according to the QA ontology. It is assumed that any time a user can query the quality evaluation of any educational resource on her choice or order some comparative evaluation (e.g., ranking list) of chosen set of resources. After the query is done, the user will be asked, according to which method ("Quality Calculator") she wants to get the evaluation. She will be offered the list of available calculators (different from each other by set of quality indicators and their relevant weights of importance, information about creators and context, in which it has been or has to be applied). If the choice is made, then the information about selected resource and its quality indicators will be queried from the remote sources by "executing" properties from the metadata layer. After that the returned data will be normalized, weighted and computed in accordance with the chosen quality calculator, and finally the user will get the required and personalized evaluation report.

The more interesting case would be if the user is not satisfied with any of the available quality calculators and wants to create the new own one to be used for quality calculations in that particular case and also in the future. The Portal allows the user to design (through the Web interface) her own quality calculator and therefore contribute to the executable knowledge creation. There are two options here depending on the user experience:

(1) The new quality calculator is based on quality indicators (executable properties, similar to ones described in Figures 6-8) already supported by the portal, i.e., there is some external source of data with the interface to it from the portal, which can be queried to get the value for each indicator. In this case the user only specifies her preferences on importance of each indicator for final quality calculation;

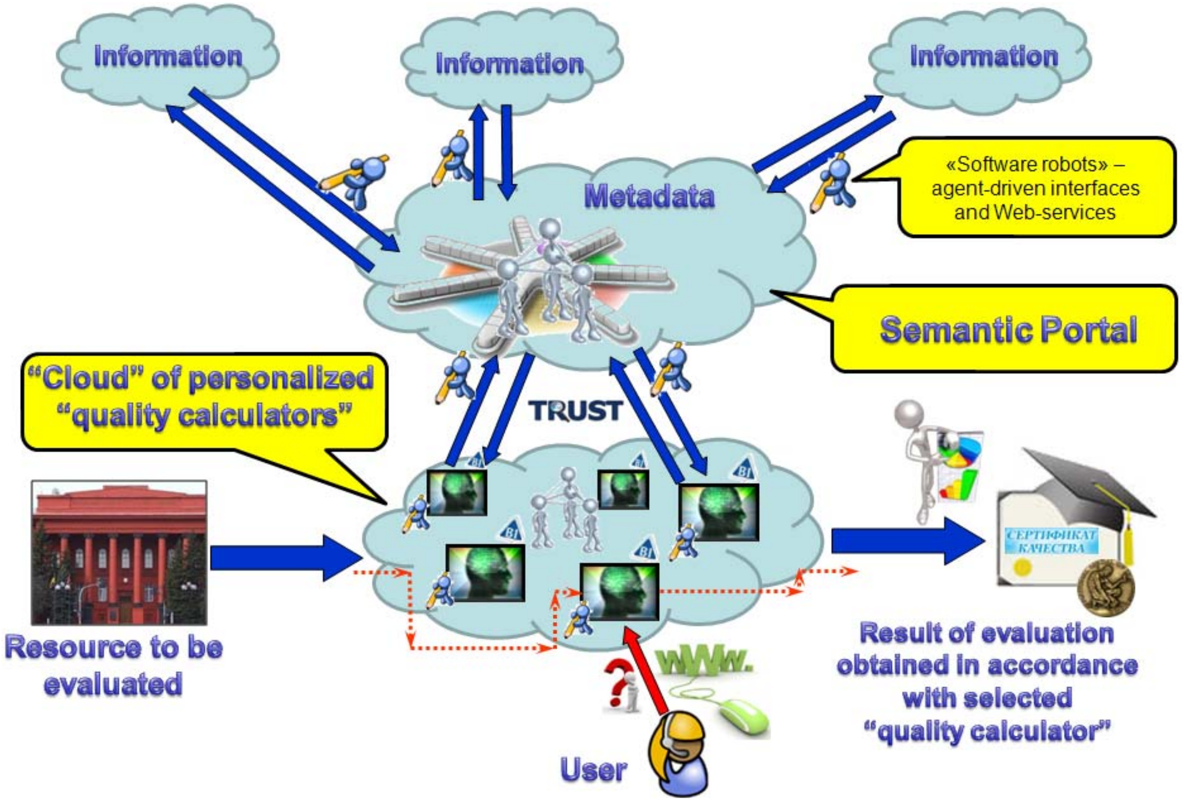


Figure 11. Executable-Knowledge-Based Architecture of the TRUST portal for quality assurance and personalized quality evaluation (a user is capable to choose or to create her own "Quality Calculator" to be applied for measuring quality of chosen resource) [32].

(2) The (advanced) user wants and is capable to add some new quality indicator (not supported by the portal yet), which means creating a new executable property and providing interface (an Ontonut) to the new source of data. Only after all the necessary and new quality indicators will be specified and remote information sources for getting values for these indicators will be available, the user may design the quality calculator itself like in the case (1).

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented the concept of Executable Knowledge, which is based on Linked Data and in addition to traditional subject-predicate-object semantic triplet model it contains also subject-predicate-query triplets (Executable Properties). We have demonstrated that data heterogeneity problem in distributed systems can be handled by the executable knowledge, which semantic (RDF) links include explicit queries to data or to (BI) services and other capabilities based on various data models and the context.

We have shown one way (named Executable Reality) on how Linked Data can be automatically processed by various BI services; and also how the results of BI processing can be requested, delivered and presented to the user through similar to the (Mobile) Mixed Reality technology interfaces.

Other executable knowledge benefits have been shown in the context of educational quality assurance and related to online quality evaluation and ranking of various academic resources. It is shown how the special Quality Assurance Portal enables executable knowledge and allows a user not only to evaluate particular academic resource based on Linked Data from external information sources, but also create her own "Quality Calculator", according to which personalized evaluations or rankings will be computed.

In the near future we are going to extend the current solutions and fully implement a powerful domain independent tool to build and execute systems on the basis of executable knowledge and to investigate new domains where such knowledge will provide an evident added value.

This paper is an extended version of conference paper [1] accepted for journal publication.

ACKNOWLEDGEMENTS

We are grateful to our international teams from UBIWARE and TRUST projects for fruitful collaboration in research and software developments; also to Tekes (Finnish Funding Agency for Industry-Driven Research) for the UBIWARE project support during 2007-2010; and to Tempus Program of the European Commission for the funding provided for the TRUST project starting from 2011.

REFERENCES

[1] Terziyan, V., and Kaykova, O., Towards "Executable Reality": Business Intelligence on Top of Linked Data, In: Proceedings of the First International Conference on Business Intelligence and Technology, September 25-30, 2011, Rome, Italy, pp. 26-33.

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

[3] Sheth, A. and Ramakrishnan, C., Semantic (Web) Technology in Action: Ontology Driven Information Systems for Search, Integration and Analysis, IEEE Data Eng. Bulletin, 26(4), 2003, pp. 40-48.

[4] Hitzler, P., Krötzsch, M. and Rudolph, S., Foundations of Semantic Web Technologies, Chapman&Hall/CRC, 2009, 455 pp.

[5] Nelson, G., Business Intelligence 2.0: Are we there yet? In: Proceedings of the SAS Global Forum 2010, Seattle, USA, 11-14 April, 2011, paper 040-2010.

[6] Domingue, J., Fensel, D., and González-Cabero, R., 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.

[7] Sell, D., Cabral, L., Motta, E., Domingue, J. and Pacheco, R., Adding Semantics to Business Intelligence, In: Proceedings of 16th International Workshop on Database and Expert Systems Applications, Copenhagen, 26 August, 2005, pp. 543 – 547.

[8] Heath, T., and Bizer, C., Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology, Morgan & Claypool, 2011, 136 pp.

[9] Khriyenko, O., and Terziyan, V., A Framework for Context-Sensitive Metadata Description, International Journal of Metadata, Semantics and Ontologies, 1(2), 2006, Inderscience Publishers, pp. 154-164.

[10] Terziyan, V., Predictive and Contextual Feature Separation for Bayesian Metanetworks, In: B. Apolloni et al. (Eds.), Proceedings of KES-2007 / WIRN-2007, Vietri sul Mare, Italy, September 12-14, Vol. III, Springer, LNAI 4694, 2007, pp. 634–644.

[11] Khriyenko, O., Terziyan, V., Similarity/Closeness-Based Resource Browser, In: J.J. Zhang (Ed.), Proceedings of the Ninth International Conference on Visualization, Imaging and Image Processing (VIIP-2009), July 13-15, 2009, Cambridge, UK, pp. 184-191.

[12] Azuma, R., A Survey of Augmented Reality, Presence: Teleoperators and Virtual Environments 6 (4), 1997, MIT Press, pp. 355-385.

[13] Ohta, Y., Tamura, H. (Eds.), Mixed Reality: Merging Real and Virtual Worlds, 1999, Springer, 418 pp.

[14] Hollerer, T., Feiner, S., Terauchi, T., Rashid, G., Hallaway, D., Exploring MARS: Developing Indoor and Outdoor User Interfaces to a Mobile Augmented Reality System, Computers & Graphics, 23, 1999, Elsevier, pp. 779-785.

[15] Henrysson, A., Ollila, M., UMAR: Ubiquitous Mobile Augmented Reality, In: Proceedings of the Third International Conference on Mobile and Ubiquitous Multimedia (MUM-2004), College Park, Maryland, USA, 2004, pp. 41-45.

[16] Mobile Mixed Reality: The Vision, Nokia Technology Insights Series, Nokia Research Center, June 2009, 4 pp., Available online in: http://research.nokia.com/files/insight/NTI_MARA_-_June_2009.pdf.

[17] Katasonov, A. and Terziyan, V., 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.

[18] Katasonov, A., Terziyan, V., Implementing Agent-Based Middleware for the Semantic Web, In: Proceedings of the Second IEEE International Conference on Semantic Computing (ICSC-2008) / International Workshop on Middleware for the Semantic Web, August 4-7, 2008, Santa Clara, USA, IEEE CS Press, pp. 504-511.

[19] UBIWARE: http://www.cs.jyu.fi/ai/OntoGroup/UBIWARE.htm, Project Web Site, Industrial Ontologies Group, 2008-2011.

[20] Gelernter, D., Mirror Worlds: or the Day Software Puts the Universe in a Shoebox...How It Will Happen and What It Will Mean, 1st ed., Oxford University Press, 1992.

[21] O'Hara, M., Tuffield, M., Shadbolt, N., Lifelogging: Privacy and Empowerment with Memories for Life, In: Identity in the Information Society, Vol. 1, No.1, Springer, pp. 155-172.

[22] Husting, P., Augmented Business Intelligence in Retail, In: Microsoft BI Collaboration and Community Blog, 3 March 2011, Available online in: http://blog.extendedresults.com/2011/03/03/augmented-business-intelligence-in-retail/ (accessed 15 June 2011).

[23] Alferes, J., Pereira, L., Przymusinska, H., Przymusinski, T., Quaresma, P., Dynamic Knowledge Representation and its Applications, In: Proceedings of the 9th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA '00), Varna, Bulgaria, September 20-23, 2000, LNCS, Vol. 1904, Springer, pp. 1-10.

[24] Alferes, J., Pereira, L., Przymusinska, H., Przymusinski, T., LUPS - A Language for Updating Logic Programs, In: Proceedings of the LPNMR'99, El Paso, Texas USA, December 2-4, 1999, LNAI, Vol. 1730, Springer, pp. 162-176.

[25] Berge, T., Hezewijk, R., Procedural and Declarative Knowledge. An Evolutionary Perspective, Theory & Psychology, 1999, Sage Publications, Vol. 9(5), pp. 605–624.

[26] Jaffri, A., Glaser, H., and Millard, I., URI Disambiguation in the Context of Linked Data, In: Proceedings of the LDOW-2008 Workshop: Linked Data on the Web, April 22, 2008, Beijing, China.

[27] Glaser, H., Jaffri, A., and Millard, I., Managing Co-reference on the Semantic Web, In: Proceedings of the LDOW-2009 Workshop: Linked Data on the Web, April 20, 2009, Madrid, Spain.

[28] Halpin. H., Hayes, P., McCusker, J., McGuinness, D., and Thompson, H., When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data, In: Proceedings of the 9th International Semantic Web Conference (ISWC 2010), Shanghai, China, November 7-11, 2010, Springer, LNCS, Vol. 6496/2010, pp. 305-320.

[29] Jaffri, A., Glaser, H., and Millard, I., URI Identity Management for Semantic Web Data Integration and Linkage, In: Proceedings of the Workshop on Scalable Semantic Web Systems, 2007, Vilamoura, Portugal, Springer.

[30] Bouquet, P., Stoermer, H., Niederee, C., and Mana, A., Entity Name System: The Backbone of an Open and Scalable Web of Data, In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008), IEEE CS Press, August 2008, pp. 554-561.

[31] Tiihonen, T., How to Create Trust, In: Tempus Information Days Helsinki, Finland, November 22, 2011, Available online in: http://www.cimo.fi/instancedata/prime_product_julkaisu/cimo/embeds/cimowwwstructure/22799_TRUST_Tempus_Timo_Tiihonen.pdf.

[32] Terziyan, V., TRUST: Towards Trust in Quality Assurance Systems. Brief Introduction of the Project Idea, In: TRUST (516935-TEMPUS-1-2011) Project Coordination Meeting, Kiev, Ukraine, October 20, 2011, Available online in: http://www.cs.jyu.fi/ai/Quality-2.ppt .

[33] Web Site of TRUST Project "Towards Trust in Quality Assurance Systems" (516935-TEMPUS-1-2011), 2011, Available Online in: http://www.dovira.eu.

[34] Klymova, M., Ontology-Based Portal for National Educational and Scientific Resources Management, In: Universities Nationwide IT Days (IT-2007), Jyvaskyla, Finland, November 1, 2007, Available online in: http://www.cs.jyu.fi/ai/OntoPortal-2007.ppt.

[35] Web Site of the Ukrainian National Ontology-Based Portal for Management of Educational and Scientific Resources, 2009-2011, Available online in: http://ailab.kture.kharkov.ua/site/index.html.

[36] Nikitin, S., Katasonov, A., and Terziyan, V., Ontonuts: Reusable Semantic Components for Multi-Agent Systems, In: R. Calinescu et al. (Eds.), Proceedings of the ICAS 2009, April 21-25, 2009, Valencia, Spain, IEEE CS Press, pp. 200-207.

[37] Minsky, M., A Framework for Representing Knowledge, In: P. Winston (ed.), The Psychology of Comp. Vision, McGraw-Hill, 1975.