Computing and Informatics, Vol. 34, 2015, 473–494

COMPARISON OF INFORMATION REPRESENTATION FORMALISMS FOR SCALABLE FILE AGNOSTIC INFORMATION INFRASTRUCTURES

Bartosz Kryza, Jacek Kitowski

AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics
Department of Computer Science
al. A. Mickiewicza 30, 30-059, Krakow, Poland
&
AGH University of Science and Technology
Academic Computer Centre Cyfronet AGH
ul. Nawojki 11, 30-950 Krakow, Poland
e-mail: {bkryza, kito}@agh.edu.pl

Abstract. In the early days of computing, files where just a natural way of storing information – which reflected the way one would file their punch cards in a cabinet drawer. Unfortunately, the requirement to fragment information into such chunks, is a huge bottleneck for the evolution of global information space that the Internet has become. The concept of file causes several problems including unnatural clustering of information, unnecessary replication of data and very expensive information discovery in distributed computing environments. The overall goal of this work is to design an architecture enabling new era in computing and networking – a computing infrastructure without the concept of file. Files are seen by many specialists as one of the main bottlenecks of modern IT systems evolution. This is mostly due to a very unnatural fragmentation of information into chunks which are easier to manage by operating systems but much more difficult for information processing tools and eventually by humans themselves.

Keywords: Information representation, file systems, distributed storage

1 INTRODUCTION

Modern IT infrastructures are the basis of several everyday activities, supporting business, scientific and entertainment aspects of human life. Although the evolution and its adoption to new areas of the IT seems unstoppable, several problems do exist at the very basis of the computing technologies applied today. Most of these problems stem from the fact that the core standards, protocols and concepts on which the modern IT infrastructures are based were designed and implemented over 30 years ago, when such ubiquity of computing devices and networks which are available today was unthinkable. One of such problems, which is very difficult to notice at first, is the concept of file. In the early days of computing, files where just a natural way of storing information – which reflected the way one would file their punch cards in a cabinet drawer. Unfortunately, the requirement to fragment information into such chunks, is a huge bottleneck for the evolution of global information space that the Internet is likely to become. The concept of file causes several problems including unnatural clustering of information, unnecessary replication of data and very expensive information discovery within Internet.

The presented research addresses this issue in order to investigate various areas of computing from information modeling, through storage technologies, security to user interfaces, in order to analyze how the future Internet could evolve once the concept of file and file system was abandoned (see Figure 1). This requires that several novel approaches must be developed for allowing to store information directly in the global information space, according to a novel information modeling paradigm, which will be stored in a specially designed distributed storage architecture, secured through advanced and distributed policy authorization system and available through context-aware user interfaces.

Most research in the area of making the existing directory based file systems more flexible can be classified into the area of semantic file systems [13], i.e. file systems where files have attached meaning. This paper sketches a vision of file systems where files can be annotated in some way, and the basic file system operation such as copy or delete do not take directory paths as arguments but the 'semantic' description of the files. The problem with these solutions is that still all the information is either fragmented or clustered into files, and the semantics deal only with meta data attached to these files in the form of some attributes. Nevertheless, these solutions are very important for our work as these approaches address important issues, mainly of how information can be found in the file based systems. One of the formal attempts at the file system implementation based on set theoretical basis is a file system using Formal Concept Analysis [12], which employs the FCA formal model of classification, neighborhood estimation and Boolean querying. A similar approach, although still bounded by the constraints of regular files, is the Logical File System project [22]. The basic role of this file system is to allow searching for files using first-order logic formulas instead of conventional directory paths. Unfortunately the use of first-order logic inference can seriously impair the scalability of the system in highly distributed settings. Until now, one major industrial attempt at abstracting

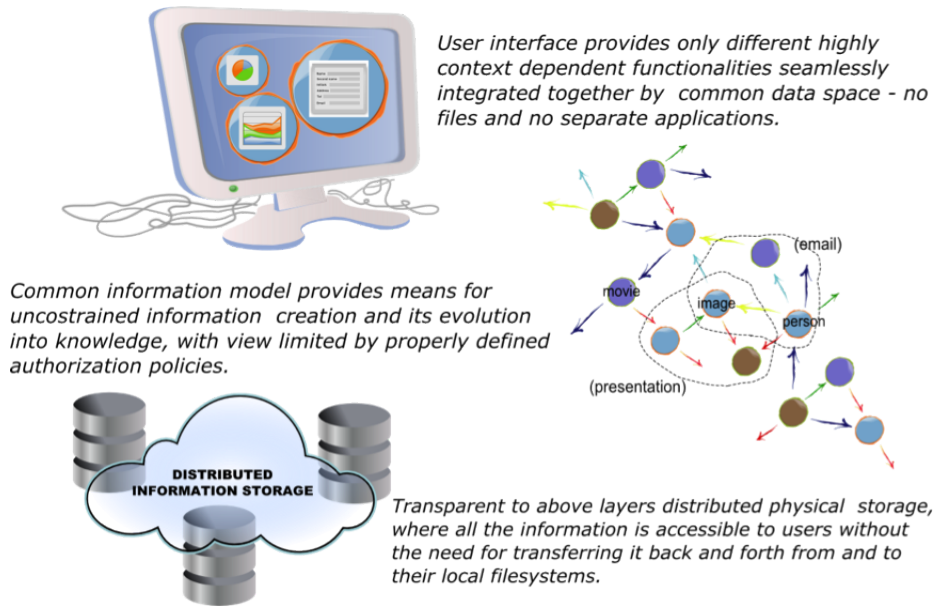

Figure 1. The overall vision of the proposed approach

the file concept from the operating system was the WinFS (Windows File System), which is a research effort from Microsoft [14]. Its basic assumption is to store all information about data in the system, including what would usually be referred to as a file in a relational database. With respect to high scalability, an interesting approach is represented by the Google BigTable system [5], which allows to store up to Peta bytes of information about URLs indexed by Google. However the information is stored in tables, columns and rows and is accessed on a key-value basis in order to be optimized for storing information about URLs and the implementation itself is based on the Google File System, thus still all the information is inherently chunked into files. With respect to more flexible user interfaces, which could naturally evolve from the proposed solution of file-less environments, several research projects have already addressed that issue, although they were still limited by the file-centered nature of current information systems. NEPOMUK [15] is a project whose goal is provision of a semantic desktop based on Semantic Web technologies to knowledge workers, by extending most popular applications with ability to process semantic annotations of data. Currently several new formalisms and technologies begin to emerge, such as technologies related with Semantic Web [4], including RDF and OWL [16] and various knowledge base solutions which allow to annotate web 'files', that is web pages [18, 19, 21]. Although created for the purpose of annotating existing information, these technologies by themselves could possibly be used to provide a basis for our vision. Additionally several formal models of information categoriza-

tion and abstraction have emerged, which can be useful in this research. One such example is Rough Set theory [24], which provides means of automatic reduction of attribute space required for generating an equivalent classification of objects, and thus could be used as an optimization form for indexing of information within the system. Furthermore, on the low level of storage device controllers, there is a trend to move from block device based interfaces (i.e. supporting file oriented systems) towards more flexible solutions such as OSD (Object-based Storage Device) [2], where instead of storing data in fixed size chunks the data can be stored in custom clusters of data along with relevant meta data. Unfortunately, most operating system level approaches still use these devices to store files, even if more efficiently [32, 30]. However, with removal of the concept of file this approach will be a significant factor along with further adoption of SSD storage [25]. In fact, Seagate has introduced recently an actual network attached object based device called Kinetic Storage [27], which provides a hardware back end for object based databases without any file system protocol access. Furthermore, in enterprise settings we can see a tendency to move away from file based databases where large amounts of information cannot be processed in satisfactory time constraints and a different approach is taken where the entire database can be stored in the memory (in-memory computing) [11] and the distributed system adheres to the "shared nothing" paradigm [31].

As we can see, there already exist several approaches and basic technologies which can support the proposed research concept. However, none of the existing solutions addresses abandoning the concept of file as a whole, including all its repercussions on the storage, operating system, application and user interface level.

2 SAMPLE USE CASE

Imagine that nowadays Anton, Danny and Mike are all researchers in the field of quantum physics. Each of them works for a different institution and has access to different libraries (both normal as well as on-line) and of course has regular Internet access. Anton regularly posts his thoughts and conclusions of his experiments on his blog. Recently he added an interesting post on how Greeneberg-Horne-Zeilinger (GHZ) state makes Bell's theorem invalid and sheds a new light on Bohr's complementarity postulate. He also included a sketch of a proof in the form of a Mathematica script. Around that time Danny made a presentation during a conference where he discussed how EPR paradox is logically invalid with respect to GHZ paradox. Most of his conclusions where in the form of bullets and diagrams on the slides of the presentation – with references to his recent yet unpublished papers. He also referenced Anton's blog and included one of the final equation from the Mathematica script by copying it manually in his presentations math editor. Some time after that Mike, preparing article for Scientific American related to the subject in matter, comes across the Anton's blog as well as Danny's presentation. As both sources contain very valid information supporting his article he referenced them both simply by adding the URL in the box in the article, adding a comment that the proof of the

Anton's idea is in the form of Mathematica script and copying also the last equation of the result into the form accepted by the journal. He also 'quotes' a couple of bullets and comments them in one of the paragraphs of his paper. In this way, Anton's equation has been actually copied into 3 different places (files). Additional copies are made as Mike sends his paper to the publisher, and then copies of the journal are distributed in PDF files. With every copy of such fact, as in this example Anton's equation, the route to the original piece of information is becoming longer and longer and eventually it is very hard to find the actual original source of the information – which is true in most cases today, when we try to find something in an Internet search engine.

Now imagine that Anton, Danny and Mike work in a distributed environment like Internet, where the concept of file is not present at all. Anton, using the unified user interface 'posts' a blog entry and extends it with a reference to a proof object he created before with the same interface but in a different – math oriented – context. Neither the blog entry or the mathematical proof are stored on his local computer – after their inception they became part of a global information space and, as set by Anton, they were accessible by anyone (as blog entries usually are). The actual structure of the information in the global information space would be more or less like this:

```
(BlogEntry:'2345123'
    (Author:#Anton)
    (Text:#Text124312312312)
    (Equation:#234562342834241123) ...)

(Person:'Anton'
    (Name:'Anton')
    (Organization:'AGH') ...)

(Text:' Text124312312312'
    (Paragraph:'EPR paradox, since it's publication in 1935...')
    (Paragraph:'Bell's inequality, presented in 1964...'))

(Equation:' 234562342834241123' ())
```

Now, using his own interface on his network computer, Mike is reading the blog. At first it seems like he is just browsing text in a text editor, but simply double clicking on a paragraph allows him to annotate the single paragraph and move a reference to it to his article, as well as reference the equation without the need to copy it at all. After completing his article, Mike submits the paper to the journal. By simply notifying the journal that an object in the information space has been created which references several textual and graphical objects he just created for the article as well as references to other objects such as contents of Anton's blog. The paper does not 'contain' any part of the Anton's blog. The journal editor receives 'references' to a dozen other publications which will be part of the next issue. Now

in his user interface he simply clicks the Publish button and a new object is created which just groups the references to objects sent to him by authors. In order to solve versioning issues, e.g. when Anton modifies the equation he published on the blog – in fact he has to create a new version of the equation as the old one is still referenced – and it is up to the owner of the referent object to decide whether to include the new version of the object or not. Of course, the authors have beforehand set up proper access rights to these objects so that the journal editor had read access to them. Now, the online subscribers of the journal are simply given read access to the object representing the new issue as well as to the referenced objects through proper delegation mechanisms in the authorization infrastructure, and the special plug-in to the interface distributed by the journal to its subscribers properly renders these objects on the screen to emulate an experience of actually reading a journal.

3 VISION AND REQUIREMENTS

The envisioned impact of the proposed approach to existing and emerging aspects of IT includes:

Seamless collaboration and social aspect. Our approach would bring collaborations between peers, such as regular computer users or even entire organizations into a completely new level. Since all the information created and modified would be stored in a distributed network, each piece of information created such as diagram or text document, would be accessible (after application of proper access rights) to anyone who might need them. Thus the improvement of the way internet is used currently would substantially improve the social evolution process in general, influencing European economy and society. The project will provide a scientific foundation of a new methodology for future IT systems organization and its exploration via proof of concept, which will provide more flexible and broad access to richer information and data according to users' privileges, better management of data due to its disambiguity, influencing highly global societal, medical and cultural challenges in a long-term perspective.

Minimal data redundancy. This approach could drastically limit the amount of data duplicated constantly while producing and transferring files. For instance a picture sent in an e-mail does not need to be copied into the user's file system while being sent to him, in fact there would be no e-mail sent as there would be even no destination file system. All that would happen is that the sender would create an e-mail 'object' and the recipient would be notified of its creation. The email context of the recipient user interface would be able to access the email object and render this message to his user on demand, without a need of creating a single new file. However this distribution would not be random, as in case of attachments sent in e-mails, but could be optimized by proper algorithms and ensure no single point of failure of such system.

No separate applications. Currently in order to perform such simple tasks as to prepare a text document, presentation or even to write an email, often the user

must use several applications in order to get access to all necessary functionality. What is worse, the information between these applications is transferred in the form of files. For instance, user creates a chart in a spreadsheet, a diagram in an SVG editor, and a text document in a word processor. Each of these actions requires opening of a new application, creation of new files, and eventually, all these files can be merged into one document. Our approach would allow developing user interfaces where the applications would only provide additional context dependent functionality, but there would be no need for switching between applications and storing the intermediate information pieces into files.

The overall vision of the proposed approach is to outline the basis for the possibility of future IT infrastructures where the need for fragmenting the information into files does not exist, but the information is available in one complex data space. Once embraced, this idea opens a whole new perspective on how the IT system can work and what new possibilities are emerging, which involves research on several issues:

Data, information and knowledge are structured into 'objects'. An object is a piece of information which can be referenced by other objects, but it is not in any way atomic nor it should be understood as 'some sort of file'. 'Objects' must be able to change dynamically as the information is updated or its context changes through references. The references are treated as simply additional attributes and the special meaning can be assigned to them by proper rules, possibly context dependent. For instance, a containment property means that one object references another by means of is-part-of relations. These however cannot be imposed in a top-down manner on the users or developers, like in case of ontologies, but should emerge naturally during the evolution of the system. Rather bottom-up approach will be used to enable the composition of more sophisticated information from different kinds of objects keeping elementary information pieces.

Object decomposition and composition. Each 'object' can be decomposed by referencing (naming) its part/parts and be composed from other objects or parts of objects. This allows to simply create new objects by describing existing objects. For instance a text document 'object' can be decomposed into chapters and sections so that each of them becomes a new entity, and can be used for 'creation' of other 'objects', thus allowing dynamic granularity of information.

Adaptable user interfaces. The operating system could actually analyze dynamically the context a user is in at any given moment, and on this basis would be able to adapt the interface presented to the user. For instance, a user creating a presentation usually needs several functionalities such as diagrams, photos, animations, text and references – all this information would be available to him naturally through the context without the need to switch to different applications, and moving results created in these applications through files to the presentation.

Security and access right management. The security – in consequence of the minimal data redundancy feature – can also reach completely new level, as this system could enforce complete control on who can access which actual piece information and how on a very fine grained level. For instance, document 'object' is created which 'contains' a copyrighted picture. Since the attachment of the picture into the document is made solely by means of adding a proper attribute to the document 'object', people who read or even modify the file, do not ever copy this picture to their file system, which in fact does not even exist.

Impose minimal possible structure on the information. The information structure that will be supported by the system must be as flexible as possible on the one hand, while on the other still should enable most of the conventional file system operations such as finding and modifying information.

No information is removed. Although not clear at first, this approach comes with a very important consequence – impossibility of removing any information. Information removal in fact generates new data, i.e. the fact that the object was removed. Of course an 'object' can be made inaccessible to users and thus virtually deleted, but due the assumed flexibility of the model actual removing of an object could have large impact on the entire information space.

Formalization of the structure of information. This is the major issue which must be answered by this project, i.e. which formalism or technology to use for the purpose of modeling information from data in such system. Although several technologies such as those related to Semantic Web exist, it is not clear whether they will be practical for such large scale environments and use cases related to it. The most essential question is how to represent a piece of information, along with related data (e.g. a piece of text, a picture or audio stream) and the attributes providing its context.

Intelligent information storage. Another critical question is what technology should be used to eventually implement a storage necessary for maintaining the distributed repositories of such information. Here the most important issues are how to store such information using existing technologies (relational databases, XML databases or other) during the first phase of experiments and how to eventually move to actual information repositories which could be 'native' to the proposed information model. Additionally, a low level approach to the distribution and discovery of information in such storage must be analyzed based on emerging technologies and standards including Solid State Disks and with local, more intelligent processing units, working both in pull and push modes.

Search through the distributed repositories. Discovery of information should be possible by virtually unlimited number of ways. Queries by object name, attributes, context dependent queries, queries with relations between objects and notification of user when some information that might be of interest to him 'emerges in the network'.

Securing the information. The security of such information will be a very challenging issue, especially due to constantly changing structure of existing information, i.e. composition and decomposition as well as creation and modification of objects attributes.

4 REVIEW OF SELECTED INFORMATION REPRESENTATION THEORIES

Most general information theories in computer science have some basic things in common. First of all, let us assume that the information system consists of objects that represent real or virtual instances of concepts. Furthermore, the objects can have certain attributes which can be assigned to sets of values. The attributes can usually have certain constraints on the domain of values which can be assigned to them (e.g. integer, string, other objects, etc.). Several challenging developments have been also made available in the recent years such as information flow theory and the theory of information algebra which provide theoretical grounds for a general fabric of distributed information system where the concept of file is no longer necessary or feasible. Although each of the discussed below novel approaches has a different focus, they provide several features necessary for a universal information representation system:

- classifications, i.e. possibility of defining a relation from objects and types
- reification i.e. allowing treating statements as first class objects (or at least allowing to treat types/attributes as first class objects)
- object attributes and properties
- provenance, i.e. enabling tracking of authorship properties on statements of the information
- locality of classifications and attributes (local logics), each user can model their knowledge from their own point of view and it is the matter of application and context logic to merge such local logics to a complete knowledge,
- open world assumption,
- mappings (infomorphisms, Chu transformations)
- flexibility of schema
- information ordering (more generic information vs more specific information)

The goal of this section is to review some classical as well as more recent advances in information representation theory.

4.1 Relational Models

One of the first formalization of information representation and structure that found worldwide practical application was the relational model introduced by Codd [7].

Before the relational model most information systems were informally modeled using hierarchical or network topologies. In fact one of the motivation for this model mentioned by Codd in his original paper was the lack of referential structure in data stored in regular files, where three data dependency categories are shown to be very difficult to achieve with file based data storage: ordering dependence, indexing dependence and access path dependence. The relational model is based on first order calculus. The formal model can be best represented using set theoretic definitions. First of all, a relation R on a family of sets S_i is a set of n-tuples whose values on i^{th} position belong to the set S_i, or more formally:

$$R \subset S_1 \times S_2 \times \cdots \times S_n$$

The S_i constitute the domains of the relation while the tuples constitute the body of the relation. All tuples must be distinct. The relation also contains a naming of the domain sets, which constitutes the header of the relation. An important notions of relations are the candidate key and primary key, where the former is a set of subset of the domains which uniquely identify the rows in the table, while the latter is a designated candidate key for given relation.

The relational model proposed by Codd also introduced formal means for normalization of the relational model enabling its effective storage in the form of tables. The so called first normal form ensures that the graph of relationships between domains are in the form of trees and that no domain constituting a primary key is complex (composed of other relations). Further normalization steps include second normal form [8], which ensures that any attribute which is not part of any candidate key is dependent on the entire primary key. This norm decreases the redundancy, and more importantly, protects the data from update anomalies related to updating a value repeated in several rows in only one leading to inconsistent data. Finally, the third normal form requires that the every non-key value is dependent on the entire key of the relation and nothing else.

It is important to mention that most existing relational databases in fact do not conform to the actual theoretical model presented by Codd, as they are based on the SQL (Standard Query Language) which deviates from the relational model in several ways such as allowing anonymous or duplicate columns (attributes), duplicate rows and handling of NULL values. From the point of view of this analysis, relational model has very strong constraints on how the data is modelled. Most importantly it is very difficult to model and efficiently store object or graph-based structures in relational databases (object-relational impedance mismatch). This has been addressed in [9], largely blaming the SQL language design decisions. The authors propose to maintain the relational model for object based systems, and to introduce more advance typing mechanisms capable of storing application level objects. However, object based models are still not perfectly suited for general information modelling, which most often, is most compatible with a graph based structure.

The main problems with relational model from the perspective of the presented research vision are twofold. First of all, relational databases have inherently poor

scalability characteristics. Secondly, the relational model, although appropriate for certain applications where data integrity is more important than scalability, are not feasible in highly distributed infrastructures with loose and dynamic schemes of information, as has been shown by the prominence of so called NoSQL database systems in the globally scalable web applications.

4.2 Pawlak's Information Systems

Pawlak's work is one earliest attempts to provide general theory of attribute based information systems [23], which combines formal definitions of the query language on both syntactic and semantic levels simultaneously. In other words, the query language is at the core of the proposed information system since it defines the subsets of objects based on their properties expressible in the query language.

In this model, an information system can be defined as a quadruple:

$$IS = \langle O, A, V, \rho \rangle,$$

where O is the set of objects, A is the set of attributes, V is a union of sets of values for each attribute and ρ is a function $\rho : X \times A \to V$. The system is complete if the function ρ is defined for each pair of $X \times A$.

The information system can be conveniently represented in the form of tables where rows represent objects, columns represent the attributes and values are assigned to respective cells. In this model, information about object $o \in O$ is simply defined as a function:

$$\rho_o : A \to V,\ \rho_o(a) = \rho(o, a),$$

i.e. a row of the tabular representation of the information system. More general information in IS is defined as a function:

$$\phi(a) \in V_a,$$

which for each attribute assigns the values in IS. This model provides natural means for classification of objects based on their attributes, which partition the object set into equivalence classes based on the attribute values. Furthermore, this model acknowledges the possibility of attribute dependency in a formal way by means of their equivalence classes, i.e. attribute a is dependent on attribute b if its equivalence class (sets of objects with the same attribute value) are a subset of the equivalence class of attribute b.

The theory defines the subsystem of system $IS = \langle O, A, V, \rho \rangle$ – a system $IS' = \langle O', A', V', \rho' \rangle$, where $O' \subset O, A' \subset A, V' \subset V, \rho' = \rho / O' \times A'$, where the / operator means a projection operation (a subset of the relation limited to the product of O and A subsets). In the practice a subsystem can be obtained by removing selected rows and columns from the tabular representation of the original system.

From our perspective one of the most important features of this theory is the connection of information systems, i.e. defined as:

$$\begin{aligned} O &= \cup_{i=1}^{k} O_i, \\ A &= \cup_{i=1}^{k} A_i, \\ V &= \cup_{i=1}^{k} V_i, \\ \rho_i &= \rho / O_i \times A_i, \\ \rho_o &= \cup_{i=1}^{k} \rho_{i_o}, \end{aligned}$$

where each of the systems is the subsystem of the combined information system. One important aspect of this operation is that it assumes that the resulting valuation function ρ is total, i.e. there are no *nil* values in the resulting information system tabular representation. In practice it means that in order to connect systems into a single IS, all subsystems must either have equal sets of objects or equal sets of attributes. This may seem as a very strong constraint limiting the practical applications, because in real applications a complete knowledge is not always available; it makes it more feasible for a formal query language.

This theory of information systems is the basis for a well known theory of rough sets [24]. In particular, an interesting improvement of the Pawlak's information systems is the extension proposed by [6], where partial ordering on the attribute set is introduced. This theory has been further extended into the area of uncertain information through the theory of rough sets. The main strength of this theory is formalization of the concept of information as the set of answers to questions in the information system and means for connecting distributed systems. However the connection of information systems is too restrictive for real world loosely coupled system, since the domains of attributes in this theory must be the same. In fact, it has been shown that information systems and classification systems are equivalent [29].

4.3 Description Logics

Description Logics (DL) is a family of knowledge representation formalisms based on logic-based semantics. The main goal of Description Logic based languages is definition of taxonomy and instance based knowledge along with properties on the objects in the knowledge base, reasoning over explicit information in order to infer implicit facts and prove knowledge consistency. One of the main attractive factors for DL is that for a large group of languages from this family decidability of subsumption can be proven and efficient algorithms can be developed.

For example, the most basic Description Logic language is the attributive language $\mathcal{AL}$, where concept descriptions can have the following form:

$$C, D \rightarrow A | \top | \bot | \neg A | C \sqcap D | \forall R.C | \exists R.\top$$

which means that the concepts can be described using respectively: simple concepts (defined by name), universal concept, empty concept, negation of simple concept, conjunction of concepts, restriction of relation value types and existence of a particular relation without restriction of the value types. In a more formal way, we can define their formal semantics using interpretations. These are defined using set representations where Δ is the domain of the interpretation of a given model, i.e. the set of all possible instances in the domain. This gives us respectively:

$$A^{\mathcal{I}}|\Delta^{\mathcal{I}}|\emptyset|\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}|C^{\mathcal{I}} \cap D^{\mathcal{I}}|x \in \Delta^{\mathcal{I}}|\forall y.(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}|x \in \Delta^{\mathcal{I}}|\exists y.(x,y) \in R^{\mathcal{I}}$$

This language can then be further extended by introducing additional constructors and thus enhancing the language expressiveness. For instance $\mathcal{ALCN}$ is a $\mathcal{AL}$ with arbitrary concept negation and number restrictions on roles. For instance the language underlying the Web Ontology Language (OWL) is $\mathcal{SHOIN}(\mathcal{D})$. This subject is discussed in detail in [1].

The main power of Description Logics comes from the possibility of formalizing conceptual domains on various levels of abstraction, flexible schemas and strong mechanisms for reasoning and verification of existing knowledge. Unfortunately, these features come with a high cost related to the current tools for performing tableaux reasoning.

4.4 Formal Concept Analysis

Formal Concept Analysis (or FCA) is a formal methodology for generating object classifications from existing information represented in the form of a binary relation:

$$\subset O \times A$$

where O is the set of objects in the domain, A is the set of attributes. In FCA such triple $\langle O, A, I\rangle$ is called the formal context. On such relation we can define so called concept forming operators $X^{\uparrow}$ (which assigns to the set of objects X a set of their common attributes) and $Y^{\downarrow}$ (which assigns to the set of attributes Y the set of objects with such attributes). Then, a formal concept in $\langle O, A, I\rangle$ can be defined as:

$$\langle X, Y\rangle : \; X^{\uparrow} = Y \; and \; Y^{\downarrow} = X, \; X \subseteq O, \; Y \subseteq A$$

i.e., all object attribute pairs of $\langle X, Y\rangle$ belong to relation I. The classification of objects stems from these naturally as a subset-superset relation between objects and attributes in concepts. We can say that concept $\langle X_1, Y_1\rangle$ is a subset of (denoted as $\leq$) $\langle X_2, Y_2\rangle$ iff $X_1 \subseteq X_2$ and $Y_2 \subseteq Y_1$. All concepts of a formal context along with a subconcept relation form a lattice such that:

$$\langle \{\langle X, Y\rangle \in 2^X \times 2^Y \mid X^{\uparrow} = Y, Y^{\downarrow} = X\}, \leq \rangle$$

Although this methodology is not a knowledge representation per se, it can significantly improve the automatic generation of classifications using information

about objects in the attribute form (such as tags, e.g. [10]). FCA can be useful for instance in advanced tag based information representation where various objects are assigned different tags (or attributes), and the concepts can be inferred automatically by means of derivation operators. Several algorithms for generation of the concept lattice from data have been developed and compared [20]. Formal Concept Analysis can be used directly to generate ontologies from attribute based data as shown in [28].

4.5 Information Flow Theory

Information Flow theory was developed by Jon Barwise and Jerry Seligman [3] as means for representing informational dependencies between systems in the form: *a's being B carries the information that c is D*. As such, the information flow theory is not explicitly an information representation system, however, it provided several important and unique features for modelling and reasoning about information within distributed systems. Information flow theory addresses directly one of the main problems with descriptions logic, i.e. the mapping between different classification. In fact, it provides an algebra of operations on classifications providing means for combining classifications over sets of objects. However, this theory does not specify how to create a particular classification, rather it defines them as triples of the form

$$A = \langle TOKENS_A, TYPES_A, \models_A \rangle,$$

where tokens are the first class objects of the modelled system. Based on that, the authors introduce the concept of infomorphism between different classifications A, B, as a pair of functions $\langle f^\vee, f^\wedge \rangle$, which map concept $\alpha \in TYPES_A$ to concept $\beta \in TYPES_B$ and a token $b \in TOKENS_B$ to a token $a \in TOKENS_A$ in a way that: $f^\vee(b) \models_A \alpha \iff b \models_B f^\wedge(\alpha)$. Infomorphism is denoted typically as $f : A \rightleftarrows B$. This gives a way for defining formal mappings which do not simply relate types to each other, but which also take into account the interdependencies between particular objects and types in 2 classifications. Furthermore, the theory provides means for combining classifications in the following way:

1. $TOKENS_{A+B} = \{\langle a, b \rangle : a \in TOKENS_A, b \in TOKENS_B\}$,
2. $TYPES_{A+B} = \{\langle 0, \alpha \rangle : \alpha \in TYPES_A\} \cup \{\langle 1, \beta \rangle : \beta \in TYPES_B\}$,
3.

$$\forall_{\langle a,b \rangle \in TOKENS_{A+B}} \langle a, b \rangle \models_{A+B} \langle 0, \alpha \rangle \iff a \models_A \alpha$$
$$\langle a, b \rangle \models_{A+B} \langle 1, \beta \rangle \iff b \models_B \beta.$$

This operation makes it possible to identify the origins of the concepts in a unified classification. Finally, we can present the most interesting concept of information flow theory which is the information channel that provides formal means for representing part-whole relationships. The information channel can be defined as an indexed family of infomorphisms $\{f : A_i \rightleftarrows C\}$, where C is called the core of the

channel and is the common codomain of these infomorphisms. Thus, the classification C consists of tokens, which can be interpreted as connections between tokens in classifications A_i. This leads to a possibility of combining local logics (for instance individual users information space) into a unified theory by means of defining infomorphisms between their classification and some common theory. The Information Flow theory provides in fact two inference rules supporting such use cases, namely f-Intro and f-Elim, supporting constraints on classifications. A constraint is defined as a pair of sets of types $\langle \Gamma, \Delta \rangle$, when:

$$\forall_{a \in TOKENS_A}(\forall_{\alpha \in TYPES_A}\ a \models_A \alpha \implies \exists_{\alpha \in \Delta}\ a \models_A \alpha),$$

and the inference rules can be defined as:

$$f - Intro \ : \ \frac{\Gamma^{-f} \vdash_A \Delta^{-f}}{\Gamma \vdash_B \Delta}$$
$$f - Elim \ : \ \frac{\Gamma^{f} \vdash_B \Delta^{f}}{\Gamma \vdash_A \Delta},$$

where Γ, Δ form a sequent.

Although the Information Flow theory is not directly applicable to information modeling, it provides significant means for providing interoperability on a semantic level between independent models.

4.6 Information Algebra

Information Algebra is a theory proposed in [17], which provides an extended information theory, not focusing on the statistical aspects of information transmission as in case of classical Shannon theory, but rather on the actual content of the information. The proposed approach is based on an information model where an information can be regarded as the set of answers to some question. Probably the most important benefit of this approach is a direct entailment from that of a partial order mapping of information, based on whether the particular question gives more generic or more specific answers. This means that the particular information space can be analysed using the lattice theory.

The core of the theory is the information algebra structure defined as:

$$(\Phi, D)$$

where Φ is a set of pieces of information forming a semigroup, and D is a set of questions or domains which forms a lattice, which includes the following operations. First of all,

$$D \times D \to D, (x, y) \to x \wedge y,$$
$$D \times D \to D, (x, y) \to x \vee y,$$

which operate on the domains of the algebra (or questions) and provide the functionality for combining domains into more specific (Meet) and more generic (Join). In fact, it is possible to define a partial order for $x, y \in D, x \leq y$, i.e. which defines a lattice on D and thus a relation of finer and coarser questions. Next operation is a combination of pieces of information

$$\Phi \times \Phi \rightarrow \Phi, (\phi, \psi) \rightarrow \phi \otimes \psi,$$

which is a semigroup meaning that a combination of information with itself gives no new information, i.e. $\phi \times \phi = \phi$. Furthermore, the operation of projection

$$\Phi \otimes D \rightarrow \Phi, (\phi, \psi) \rightarrow \phi^{\downarrow x} \text{ when } x \leq d(\phi)$$

provides means for focusing a piece of information to a particular domain (question), which is similar to a select query in the relational database.

In [17] authors discuss how information algebra can represent various systems such as proposition or predicate calculus, boolean algebra or even relational systems.

A more interesting issue is the one proposed by the authors' context system $(L, M, \models)$ where L is a language (or set of possible sentences), M is a model and the relation defines binary relation between language and model $\models \subseteq L \times M$, which is equivalent to the classification in information flow theory and also allows definition of operations similar to the infomorphisms.

The main strength of this theory from the point of view of the proposed research is a high degree of abstraction over modelling domains thus allowing formal representation and analysis of various information structures along with their intended meaning. Although the theory does not provide any specific means for information modelling and representation, it gives an important tool for analyzing and measuring information content, and formalises the operations of information projection and combination independently of a particular domain.

4.7 Summary

The purpose of the presented review was to identify recent efforts in information modelling and compare them against classical methods. The main features of the proposed model include: scalability, flexibility, functional dependencies between entities and federation (globally unique entity identification).

Starting from the mostly predominant relational model, the main drawback from the point of view of the proposed research is a lack of modelling of hierarchical entity structures (e.g. inheritance), and these have to be introduced separately through nested relations [26]. This makes the relational model (and in general, any value-based model), not very interesting from the point of view of this research. However, some concepts from the relational model such as functional dependencies remain useful in any information modelling paradigm. Some existing approaches propose even introduction of intensional knowledge directly in the RDBMS systems [33].

A little more interesting is the Pawlak's Information System theory which is an abstraction of the relational model with formal query language semantics. The theory also enables information distribution. A more recent and common information modelling is the family of standards such as RDF, RDFS, OWL and OWL2 all based on various subsets of Description Logics. The main strength of these languages is (depending on the particular variant) existence of sound and complete inference algorithms enabling automatic information consistency verification and addition of implicit information. The main problems with DL based languages are relatively expensive inference algorithms and lack of native support for meta-modelling (i.e. adding custom attributes and relationships between concepts and relations).

Other advances also exist which have not focused on information representation, but more generally, on what is information and how it can be theoretically approached, without considering a particular representation. One approach is the use of lattice based frameworks such as presented Formal Concept Analysis, which enables the automatic discovery of classes and relations in attribute-based information, providing a very powerful mathematical framework. Even more general approaches such as information algebra and information flow theory provide an insight into what should be considered as information, how can it be composed and decomposed and communicated.

5 CONCLUSIONS

Existing trends in the way internet is being used show clearly that the amount of content including textual information, multimedia files and raw binary data will keep increasing and we are actually at the beginning of the road to the world where all information is digitized and there is still enormous amount of data to be added into the pool. The two main bottlenecks which can slow down this process include the technology related to a distributed storage of information and the lack of information structure which could enable proper searching the information through in order to discover relevant data.

In this paper we have discussed a high level vision of a novel approach to organization of IT infrastructure – one that abandons the concept of file. The concept of file and file system that played a crucial role in the computing so far, is becoming in our opinion a major bottleneck in terms of flexibility, redundancy and scalability. We propose that basing on several novel technologies and information theories available today should be used to develop a universal model for information representation. Of course it is not possible to create a formalism which can satisfy both the high level of flexibility with the high level of expressiveness. However, the proposed vision assumes rather creation of an underlying fabric which provides means for adding expressiveness appropriate for specific applications. This is in our opinion why the existing logic based frameworks for information representation such as ontologies cannot reach a widespread adoption – they imply a certain level of expressiveness, which can be either too high or too low for particular use cases.

Also for this reason, there has been such increased popularity of various NoSQL or graph distributed databases which impose very little structure on information and allow expressiveness to be added in the application layer.

We have presented here several information representation theoretical systems, some complementary to each other and some overlapping, however they all provide a significant insight into the basic possibilities and requirements of such approach:

- There are no files – not in the storage, middle ware, operating system or user interface layers. Of course, at the prototype stage such approach would be very expensive in order to remove files completely from existing operating systems which use files even for communication with hardware devices.
- Documents, emails, images, movies, web pages and all other concepts, which are in practice today synonyms for files, in our architecture are only manifestations/renderings of interconnected groups of objects shown to the user in a context dependent way. In order to enable integration with existing environments, we envisage development of a file system plugin which could present the object as files (e.g. through a fuse extension in Unix systems).
- Data and meta data exist at the same level – for instance there is no difference between the 'Image' object and the object describing its author or authorization policy – we do not plan to introduce a meta data mechanism such as Dublin Core or even Semantic Web.
- Data and information replication should be controlled by the middleware – it is not necessary for users to copy and store the information for either security or efficiency reasons. As a consequence, data redundancy can be optimized by the middleware.
- The proposed approach inherently supports the ubiquitous computing paradigm – there is no 'Load document', 'Save document' operations. It is possible to work on a laptop, then literally just shut it down and switch to pocket PC or mobile phone and all the changes will be seamlessly available there, of course, assuming network access is omnipresent.
- Security, especially authorization is intertwined within the global information space along with information itself – i.e. security assertions (and any 'annotations' for that matter) are first class objects in the infrastructure.

In our opinion such file-less computing disruptive change could enable a significant improvement in all kinds of scenarios where data integration, replication, security and scalability are a major issue.

The future work will include development of the information storage model satisfying as much as possible the defined requirements and implementation of a prototype proving the feasibility of such approach.

Acknowledgments

This research has been funded by Polish National Science Centre Grant "File-less architecture of large scale distributed information systems" under number: DEC-2012/05/N/ST6/03463.

REFERENCES

[1] Artale, A.—Calvanese, D.—Kontchakov, R.—Zakharyaschev, M.: The DL-Lite Family and Relations. J. Artif. Intell. Res. (JAIR), Vol. 36, 2009, pp. 1–69.

[2] Bandulet, C.: Object-Based Storage Devices. 2007, `http://www.oracle.com/technetwork/server-storage/solaris/osd-142183.html`.

[3] Barwise, J.—Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.

[4] Berners-Lee, T.—Hendler, J.—Lassila, O.: The Semantic Web. Scientific American, Vol. 284, 2001, No. 5, pp. 34–43.

[5] Chang, F.—Dean, J.—Ghemawat, S.—Hsieh, W. C.—Wallach, D. A.—Burrows, M.—Chandra, T.—Fikes, A.—Gruber, R. E.: Bigtable: A Distributed Storage System for Structured Data. Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06), USENIX Association, Berkeley, CA, USA, Vol. 7, 2006, pp. 205–219.

[6] Cirulis, J.: Knowledge Representation in Extended Pawlak's Information Systems: Algebraic Aspects. Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems (FoIKS '02), Springer-Verlag, London, UK, 2002, pp. 250–267.

[7] Codd, E. F.: A Relational Model of Data for Large Shared Data Banks. Commun. ACM, Vol. 13, 1970, No. 6, pp. 377–387.

[8] Codd, E. F.: Further Normalization of the Data Base Relational Model. IBM Research Report RJ909, San Jose, California, 1971.

[9] Darwen, H.—Date, C. J.: The Third Manifesto. SIGMOD Record, Vol. 24, 1995, No. 1, pp. 39–49.

[10] Eklund, P.—Goodall, P.—Wray, T.: Information Retrieval and Social Tagging for Digital Libraries Using Formal Concept Analysis. 2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010, pp. 1–6.

[11] Farber, F.—Mathis, C.—Culp, D. D.—Kleis, W.—Schaffner, J.: An In-Memory Database System for Multi-Tenant Applications. In: Härder, T., Lehner, W., Mitschang, B., Schöning, H., Schwarz, H. (Eds.): Datenbanksysteme für Business, Technologie und Web (BTW), 14. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), 2.–4. 3. 2011 in Kaiserslautern, Germany, 2011, pp. 650–666.

[12] Ferre, S.—Ridoux, O.: A File System Based on Concept Analysis. International Conference Rules and Objects in Databases, LNCS, Springer, Vol. 1861, 2000, pp. 1033–1047.

[13] Gifford, D. K.—Jouvelot, P.—Sheldon, M. A.—O'Toole, J. W. Jr.: Semantic File Systems. SIGOPS Oper. Syst. Rev., Vol. 25, 1991, No. 5, pp. 16–25.

[14] Grimes, R.: Code Name WinFS: Revolutionary File Storage System Lets Users Search and Manage Files Based on Content. MSDN Magazine, Vol. 19, 2004, No. 1.

[15] Groza, T.—Handschuh, S.—Moeller, K.—Grimnes, G.—Sauermann, L.—Minack, E.—Mesnage, C.—Jazayeri, M.—Reif, G.—Gudjonsdottir, R.: The Nepomuk Project – On the Way to the Social Semantic Desktop. In: Pellegrini, T., Schaffert, S. (Eds.): Proceedings of I-Semantics '07, 2007, JUCS, pp. 201–211.

[16] Horrocks, I.—Patel-Schneider, P. F.—van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, 2003, No. 1, pp. 7–26.

[17] Kohlas, J.—Schneuwly, C.: Information Algebra. In: Sommaruga, G. (Ed.): Formal Theories of Information. Lecture Notes in Computer Science, Springer, Berlin Heidelberg, Vol. 5363, 2009, pp. 95–127.

[18] Kryza, B.—Pieczykolan, J.—Kitowski, J.: Grid Organizational Memory: A Versatile Solution for Ontology Management in the Grid. Second IEEE International Conference on e-Science and Grid Computing (e-Science '06), 2006, pp. 16.

[19] Kryza, B.—Słota, R.—Majewska, M.—Pieczykolan, J.—Kitowski, J.: Grid Organizational Memory-Provision of a High-Level Grid Abstraction Layer Supported by Ontology Alignment. Future Gener. Comput. Syst., Vol. 23, 2007, No. 3, pp. 348–358.

[20] Kuznetsov, S. O.—Obiedkov, S. A.: Algorithms for the Construction of Concept Lattices and Their Diagram Graphs. Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '01), Springer-Verlag, London, UK, 2001, pp. 289–300.

[21] Myłka, A.—Myłka, A.—Kryza, B.—Kitowski, J.: Integration of Heterogeneous Data Sources in an Ontological Knowledge Base. Computing and Informatics, Vol. 31, 2012, No. 1, pp. 189–223.

[22] Padioleau, Y.—Ridoux, O.: A Logic File System. Proceedings of the General Track: 2003 USENIX Annual Technical Conference, San Antonio, Texas, USA, 2003, pp. 99–112.

[23] Pawlak, Z.: Information Systems Theoretical Foundations. Inf. Syst., Vol. 6, 1981, No. 3, pp. 205–218.

[24] Pawlak, Z.: Rough Set Approach to Knowledge-Based Decision Support. European Journal of Operational Research, Vol. 99, 1995, pp. 48–57.

[25] Rajimwale, A.—Prabhakaran, V.—Davis, J. D.: Block Management in Solid-State Devices. Proceedings of the 2009 Conference on USENIX Annual Technical Conference (USENIX '09), USENIX Association, Berkeley, CA, USA, 2009, pp. 21–21.

[26] Roth, M. A.—Korth, H. F.—Silberschatz, A.: Extended Algebra and Calculus for Nested Relational Databases. ACM Trans. Database Syst., Vol. 13, 1988, No. 4, pp. 389–417.

[27] Seagate Technology LLC. The Seagate Kinetic Open Storage Vision, 2013. `http://www.seagate.com/tech-insights/kinetic-vision-how-seagate-new-developer-tools-meets-the-needs-of-cloud-storage-platforms-master-ti/`.

[28] Sertkaya, B.: A Survey on How Description Logic Ontologies Benefit from Formal Concept Analysis. CoRR abs/1107.2822, 2011.

[29] Skowron, A.—Stepaniuk, J.—Peters, J. F.: Rough Sets and Infomorphisms: Towards Approximation of Relations in Distributed Environments. Fundam. Inform., Vol. 54, 2003, No. 2-3, pp. 263–277.

[30] Stender, J.—Hoggqvist, M.—Kolbeck, B.: Loosely Time-Synchronized Snapshots in Object-Based File Systems. In IPCCC, 2010, IEEE, pp. 188–197.

[31] Stonebraker, M.: The Case for Shared Nothing. IEEE Database Eng. Bull., Vol. 9, 1986, No. 1, pp. 4–9.

[32] Wang, F.—Brandt, S. A.—Miller, E. L.—Long, D. D. E.: OBFS: A File System for Object-Based Storage Devices. Proceedings of the 21st IEEE/12th NASA Goddard Conference on Mass Storage Systems and Technologies, College Park, MD, 2004, pp. 283–300.

[33] Wojnicki, I.: Jelly Views: Extending Relational Database Systems Toward Deductive Database Systems. Computer Science, Vol. 6, 2013, No. 5, pp. 95–112.

Bartosz Kryza is a researcher and developer at the Department of Computer Science of AGH University of Science and Technology in Cracow. He has participated in several EU-IST projects as task or WP leader, including FP5 CrossGrid, FP5 Pellucid, FP5 MAGIC (during research internship in France), FP6 K-Wf Grid, FP6 GREDIA and FP7 PRACE and PaaSage. His main areas of interest are at the convergence of Grid systems and semantic technologies, SOA architectures and virtual organizations, distributed data management and P2P technologies. He is the author or co-author of about 50 research papers published in international journals or conference proceedings.

Jacek Kitowski (Full Professor of Computer Science) graduated in 1973 at Electrical Department of the AGH University of Science and Technology in Krakow (AGH-UST, Poland). He received his Ph.D. in 1978 and Dr.Sc. in 1991 in computer science from the same university. He is the Head of Computer Systems Group at the Department of Computer Science of the AGH University of Science and Technology in Cracow, Poland, and senior researcher at the Academic Computer Centre CYFRONET AGH, being responsible for developing high-performance systems and grid environments. He is the author or co-author of about 200 scientific papers. His topics of interest include large-scale computations, multiprocessor architectures, parallel/distributed computing, Grid services and Cloud computing, SOA systems, knowledge engineering and semantic technologies. He participates in program committees of many conferences, and was/is involved in many international and national projects, like EU funded Crossgrid, Pellucid, int.edu.grid, K-WfGrid, Gredia, gSLM and EDA EU- SAS. He is Polish expert (nominated by the Ministry of Science and Higher Education) in EU Program Committee e-Infrastructures (EU Unit F3 Research Infrastructures) and Director of PL-Grid Consortium coordinating the PL-Grid and PLGrid PLUS projects co-funded by the European Regional Development Fund as part of the Innovative Economy Program (National Grid Initiative, Polish NGI), closely cooperating with EGI.eu and EGI InSPIRE. He is a Member of the Interfaculty Commission of Technical Sciences of the Polish Academy of Arts and Sciences (PAU) and of the Computational Science Section of the Polish Academy of Sciences (PAN), Committee on Informatics. as well as the Editor-in-chief of the Computer Science Journal (published by AGH-UST).