Project Idea

Personal Knowledge System

Date	3 Feb 2015
Author	Robert Hoff
Contact	hoff.rj@gmail.com
Status	DRAFT
Please note	Shared in confidence, do not disclose or distribute

1 Summary

We are motivated by insight towards and technology opportunity in defining a unified and consolidated store for personal data. The background thinking is described in Section 2.1, we note some generic systems already in use in Table 1. Evernote highlights commercial opportunity, but is limited in its recognition of data-types, particularly files, and its tag-based organisation.

Chapter 4 provides background on data-structures and design of existing systems, and gives context to relaxing the restriction on data-types and suggests graphs (defined in 4.2) as a structural candidate. Wikis already use graphs, as in Figure 2, to provide multiple pathways and keyword-associations as efficient aids in information retrieval.

Graphing our data as a network of weighted nodes, and emphasising a visual display forms the main conceptualisation of this idea. Chapter 3 presents a tour of the product and features, with the visual immersion as a central way of traversing and comprehending one's information. A particular goal is to combine wiki-like content with support for files, and alleviate our dependency on the file-hierarchy. Textual compositions and list-views are integral as in Figure 4 and Figure 5.

The timing of this idea is supported by current technology. There is an emergence of graph-based databases, Neo4j is leading and open-source, that match and inspire our definition. The capability of latest browsers and html specifications provide generality and UI capability to carry the product. Dropbox and related open-source products have made infrastructure available to make data simultaneously device-independent and locally accessible.

2 Motivation

2.1 Information Management

Considering the study of *Personal Information Management*, that which enables us to fulfil our duties as individual, employee, student, friend or associate, involves enduring interaction on a scope of data that exists over a variety of services and devices; sending emails, writing reports and checking schedules are familiar tasks in today's information society.

Fragmentation in this context is a leading problem that incur costs related to acquisition, retrieval and mental workload; effecting missed opportunities, redundancy and data loss. In step with an intricate digital landscape there is associated burden of archiving and finding data. Locating a photograph that is either on the digital camera, somewhere on the file-system, on Facebook or elsewhere is frustrating and accumulates significant effort.

This suggests a need for unification, which is also supported in observation of user habits. There is a subtle point that data is unified in the *expression* of our habits, on the idea that whether personal information is on one system or many, fragmented or otherwise, it can always be modelled as a single complex. The complex exists, as partitioned or overlapping sets and between non-cooperating or cooperating agents, as a collection of access points, with many data types, methods of retrieval, structure and functionality.

To contend with this complex whole, which psychologically is often seen as the same thing, it is shown[1] that individuals already have a tendency to simulate unification through available technologies, often centering around one or more of the systems in Table 1. These have ability to be individually construed to act as reference and gateway to the rest of our digital world.

Information system	Data diversity (types)	Data structure	Data retrieval
File system	Files	Hierarchies	Traversal
Email manager	Emails, text, files	Flat, categorisations	Search, most relevant on top
Personal wiki	Text, images, embedded files	Network	Traversal, word-associations, search
Evernote	Text, images, files	Partitioned domains, tag categorisations	word-associations, search

Table 1 Comparison of information systems

[1] Includes; Bernstein M. et al. *How and Why, Information Eludes Our Personal Information Management Tools*, ACM Journal (Sep 2008)

2.2 Trends and Products

The quest to create a unified system in this vision is not entirely novel and has been attempted variously during the history of computing. A review article by ACM[2] describe attempts that exhibit, in some cases introduced, hypertext linking, object-mapping and spacial layouts. Even two famous papers[3][4] from ancient history predicted an intimate coupling of computer and man, towards a kind of auxiliary brain.

Evernote actually uses the brain-analogy in their mission statement, and they are interesting as a mainstream and commercial success with this explicit intent. It is a confirmation that commercial traction for this class is possible, where earlier a combination of need and technology ability was not supportive of this opportunity. Particularly the information demands today are higher and more difficult than ever, so the introduction of these tools give individuals significant competitive edge. Technology ability before was simpler, many of the features from the review paper moved into open technologies such as the wiki, and it was harder to compete with the file-system, despite drawbacks has reliable and ubiquitous support.

There are general trends in product development that show unification providing compelling simplification. Dropbox introduced file-consistency between devices, which taken together with a wi-fi enabled camera and software for social sharing may seem to be a good attempt at solving the problem mentioned in Section 2.1 .

2.3 Limitations

A motivational driver is the notable lack of fluency in Evernote with file processing, because it hasn't been specifically conditioned for it and there is missing cooperation with the local system. Dropbox solves this succinctly by imitating the local drive, but is limited to files only. However there are no technical barriers to moving files into browser based interfaces.

An inherent problem with file-systems is their hierarchical nature, with roots in computer architecture they are designed for system function, on this premise it is reasonable to believe they are not ideally conditioned for user psychology, as illustrated below.

---

[2] Stephen Davis. *Still building the memex*, Communications of the ACM (Feb 2011)
[3] Vannevar Bush *As we may think*, Atlantic Monthly (Jul 1945)
[4] Licklider JCR. *Man-Computer Symbiosis,* Transactions on Human Factors in Electronics (Mar 1960)

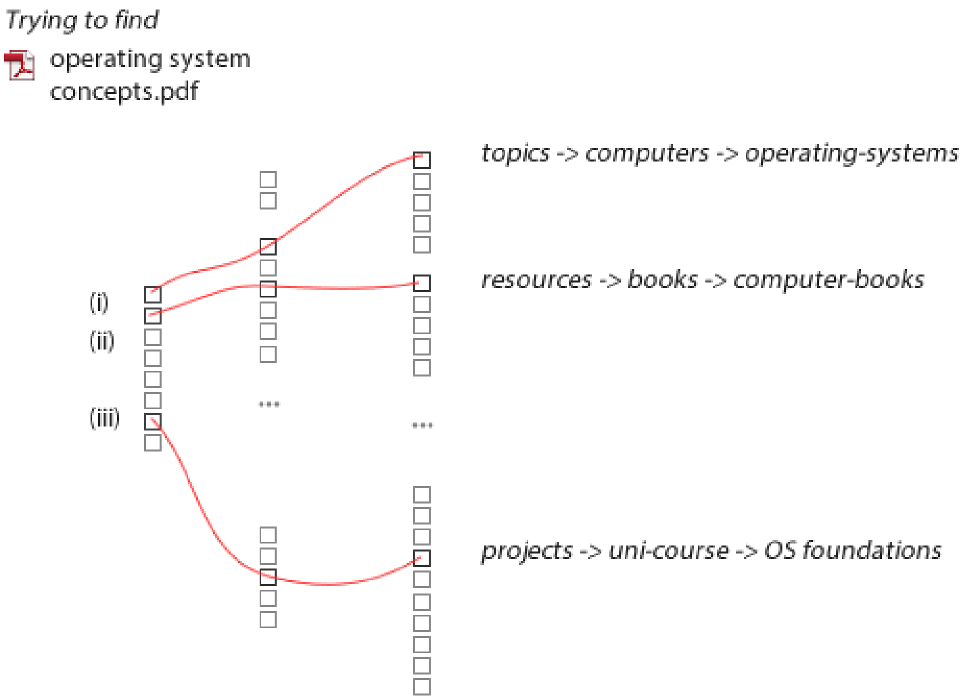

Figure 1 Competing logical schemes impose a time-cost if a user performs inaccurate traversals in search of a given item.

Figure 1 shows erroneous traversals that may arise when looking for a given item. This kind of problem is familiar, as volume of information grows in the hierarchical approach it becomes harder to maintain logical consistency.

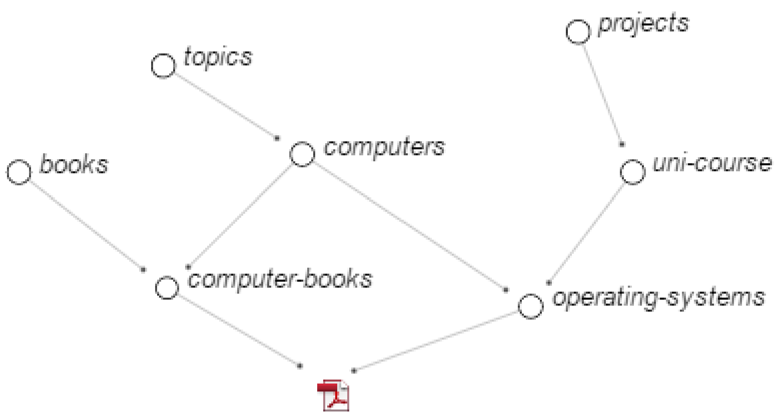

Figure 2 An alternative scheme with interlinking folders

If folders were linked together as in Figure 2 it would provide better protection against erroneous traversal. Wikis are organised like this and give indication for using networks as a better underlying structure.

3 Product

We aim to investigate the utility of a system where we relax the limitations on the number of data types (as defined in Section 4.3) and using a graph-based structure. The graph-approach seems appealing because of a general nature (Section 4.2) and can be implemented by emerging technology, particularly the open-source graph-database Neo4j (Section 4.4).

type	**synonym**	
page	node	A structural item that forms the skeleton of our system, may accept keyword to aid traversal and search, and may accept textual content
person		Any individual, friend, associate, professional, famous, etc
bookmark	url	An internet reference
webpage		A local copy of web content
code		a programming code snippet (java, python, html, etc)
document		word, pdf, open-office, etc
image	photo	
album		A collection of images
account		A place where credentials are registered, commercial services, etc
organisation	institution, business	
event		
book	publication, paper	
quote		
video		local video-files, references to online videos
contact		Such as address, phone nr, email, relating to a person or organisation
business-card		
location	place	
email		A copy of email subject and texts
task		assigned-to, due-by

Table 2 A candidate set of data-types

Table 2 gives a set of example definitions that the system may be composed of, illustrative of common types that we interact with daily in a variety of contexts.

3.1 Universe of Data

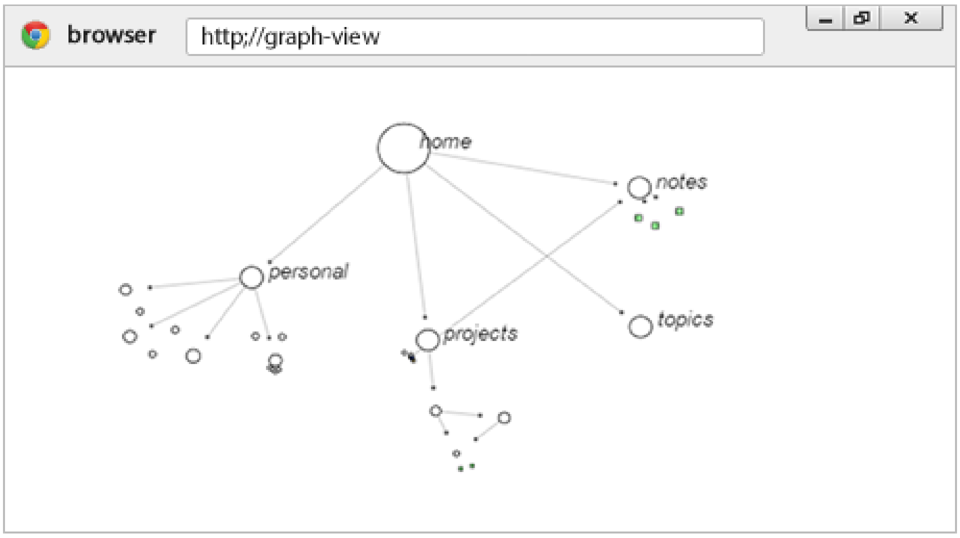

Figure 3 Graphing personal data as a network of interconnected nodes

We imagine a product that caters for a larger variety of data-types, such as in Table 2, and uses wiki-like associations to graph our data into a network. The visual result would be like Figure 3, showing a high-level view populated with a person's personal information. Offering such a visual interface would aid in traversal and interaction, and provide editing features for drawing relationships and organising the data.

The nodes in Figure 3 are shown to different weights derived from their position in the network (a node that is centrally connected, contains new data and is frequently visited will appear larger). The idea of weight is something Google uses in their Internet search algorithms (page rank), an idea which in this model is transferable to personal scope.

Considering the data as a landscape or Universe, with data-points of different strength, we can offer the ability to view it at different resolutions. Such as Figure 6 and Figure 7 to zoom in on point of interest, and in Figure 8 to accept keywords that sends the user to a close-up of a matching node.

Together with the visual conception, Figure 4 suggests that points in the graph can be viewed as structured content similar to a webpage, a presentation that can be derived from the content of the given and surrounding nodes (such as the example in Section 4.5). To duel as a file-system, Figure 5 shows a list of files displayed at a particular focus.

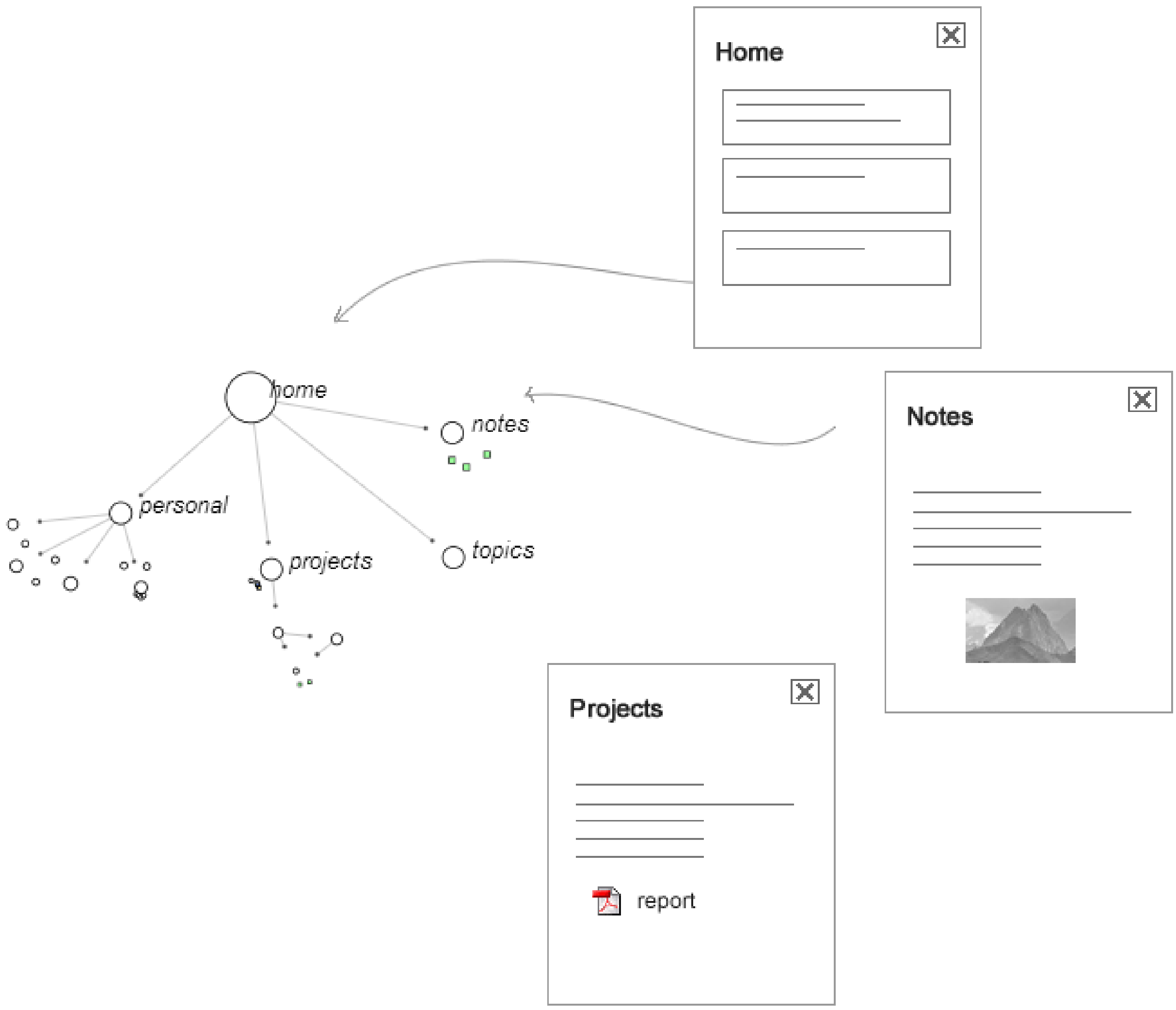

Figure 4 Points in the graph rendered as pages (consider Figure 20 for detailed case).

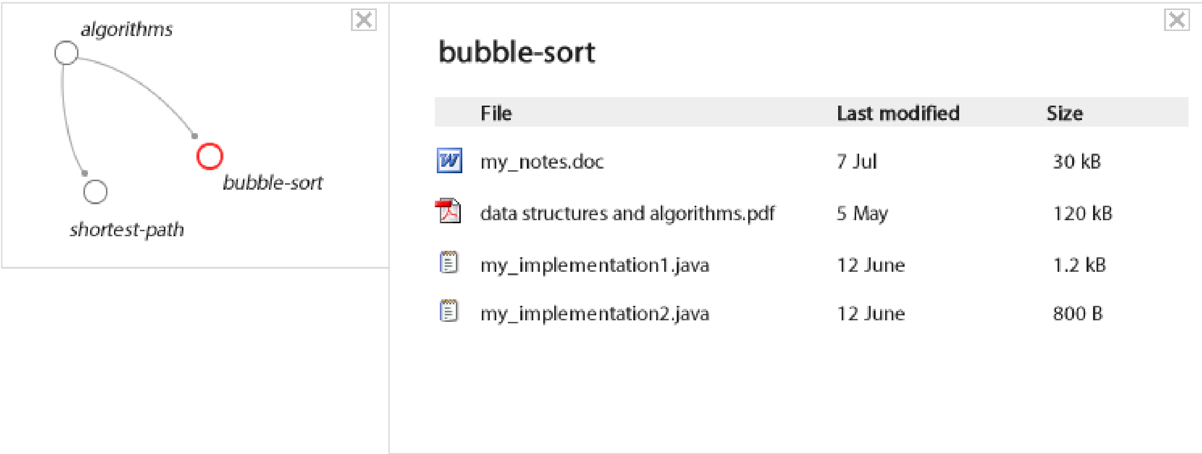

Figure 5 A sub-section of the graph with associated files

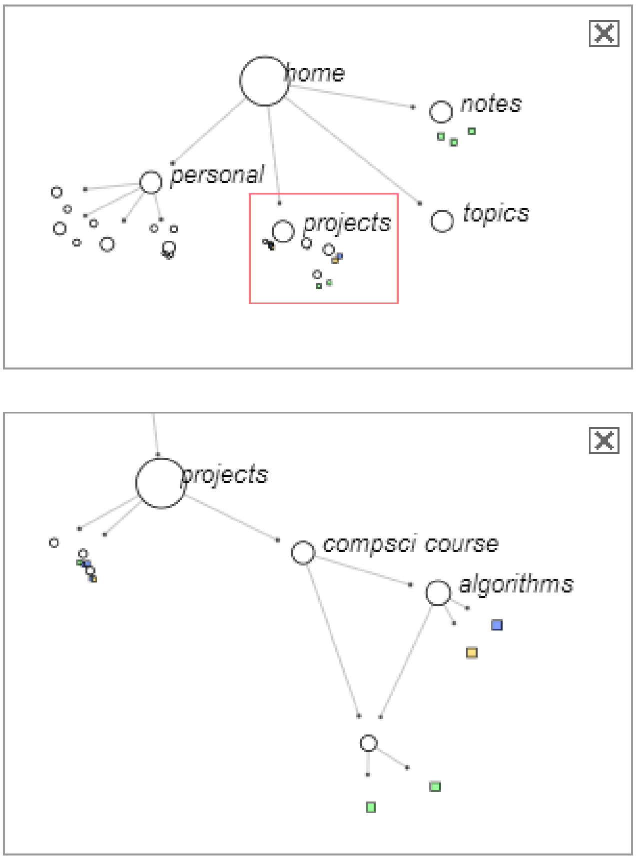

Figure 6 A closeup of a sub-section of the graph, with more detail coming into view.

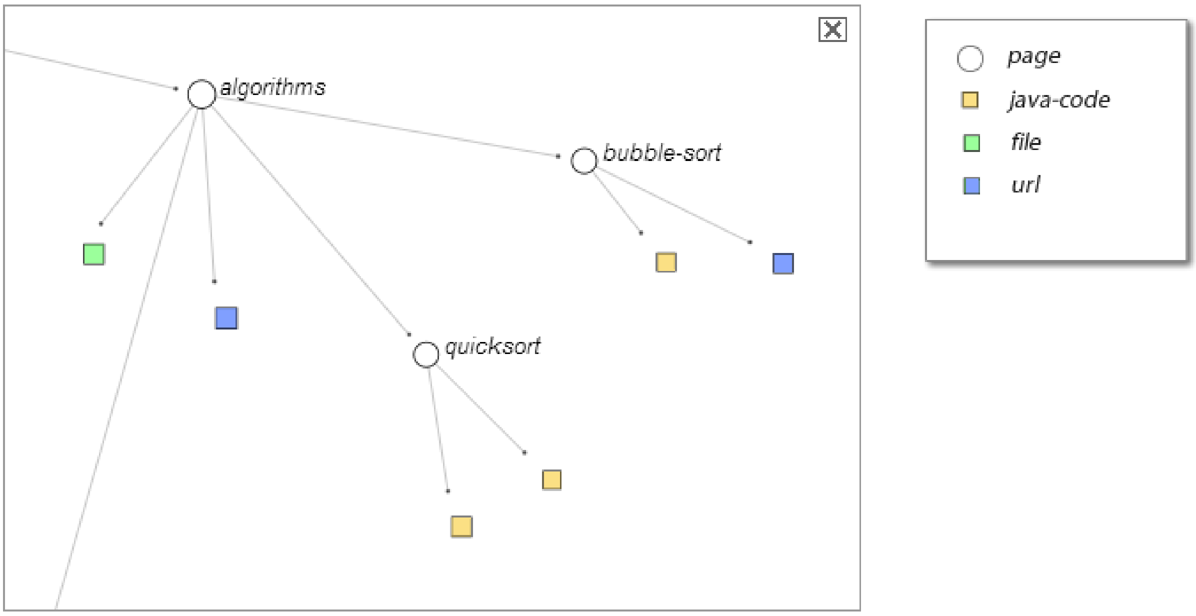

Figure 7 Different data-types distinguished visually

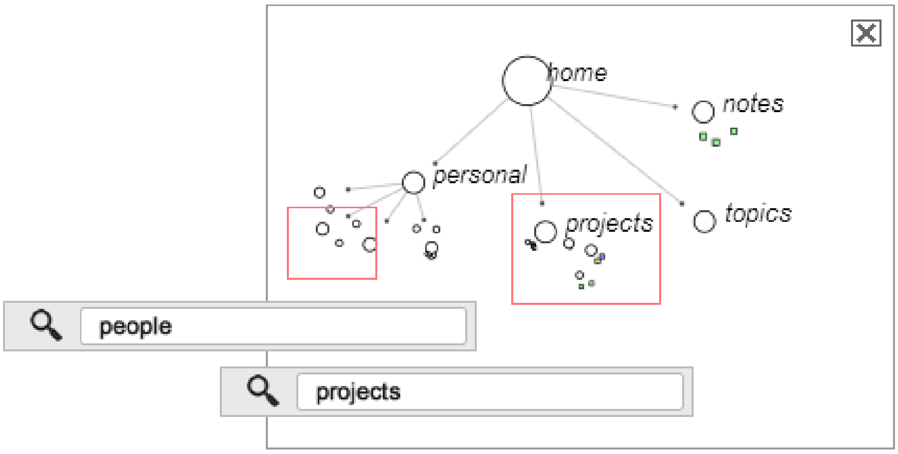

Figure 8 word-associations taking the user to different entry-points in the graph

3.2 Feature Summary

Graph-based and visual	Traversal by zooming and panning. Matching key-words to particular entry-points (Figure 8)
	Node-weight derived from centrality, content and user-history.
Many object types, and support for files	Candidate set of objects as in Table 2
	Files dragged and dropped from local system, file-processing supported by local plugin
Information retrieval	Support searches based on type, date, and in position in graph.
Browser-based	Carries all needed functionality
Interaction with third-party applications	Copying in emails from client, and sending emails on its behalf
	Able to cut in web-content and URLs
	Collect and view data on smartphone

4 Background

4.1 Generic Model

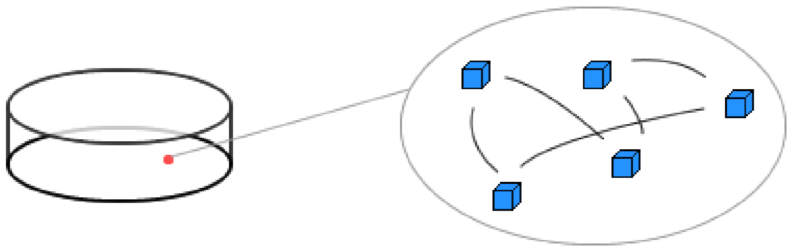

Figure 9 A given information system modelled as a structured collection of finite objects.

4.2 Graph Nature

All the different software systems in Table 1 may be described by a similar generic model, illustrated in Figure 9, mathematically a directed graph, $G = (V, E)$. Where *V*, the vertices, are a finite set of data-objects, and *E*, edges, are pair-wise relationships between them.

Within such a representation each of the systems would have particular definitions and restrictions to what the vertices and edges *are,* and to what relationships that are permitted. To illustrate, file-systems can be defined by two types of objects, files and folders, and one type of relationship, that we can call *belongs-to* or *parent-child*. Additionally, edges are only permitted from folder-to-folder or folder-to-file, and must collectively be acyclic and connected (defining a tree).

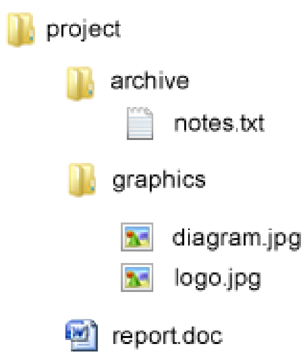

Figure 10 A file system

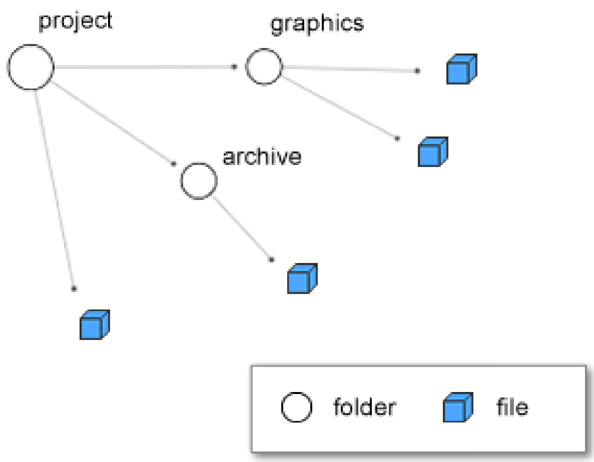

Figure 11 Alternative representation of the files in Figure 10, modelled as a graph

As claimed we can also model the other systems in Table 1 as graphs. A wiki can be described as *pages* with interconnecting links as in Figure 12. Evernote is structured around what they call *notes* and *tags,* Figure 13.

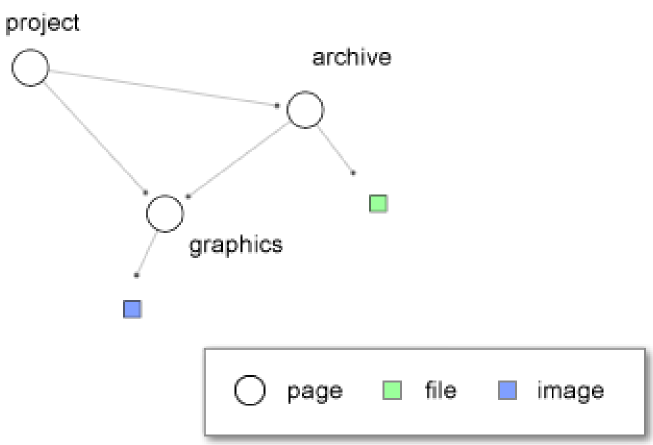

Figure 12 A wiki modelled as a graph. The main content of wikis are pages of text (white circles) with hyperlinks between them (the edges). Pages may also include files or images.

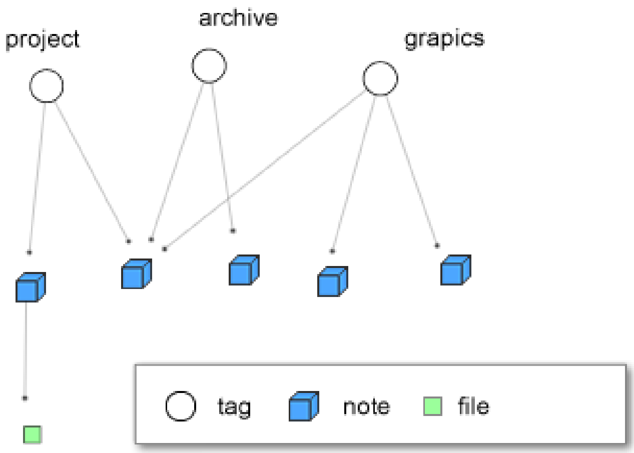

Figure 13 Evernote modelled as a graph. The main body of content on Evernote are notes, that are sectioned with tags (word-associations). Each tag returns a subset of the overall data and may overlap, so filtering may be done on one or more. Embedding files into notes is supported, shown here with one instance.

4.3 Data Definition

Modelling data-systems as graphs has recently been gaining mainstream adoption. Neo4j is the leading system and open source, it describes its data precisely as $G = (V, E)$. In Neo4j the vertices are called *nodes* and the edges are called *relationships.*

Virtually all database systems, including seasoned relational databases, Neo4j itself and other modern systems, describe their data-objects (traditionally called records) as sets of key-value pairs (fields), and denoted as a type.

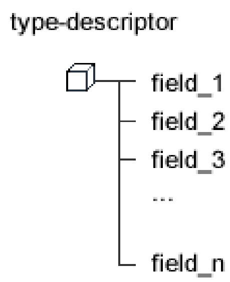

Figure 14 Neo4j and other database systems define their records, or *nodes* in Neo4j, as assigned types with corresponding fields, as the example in Figure 15.

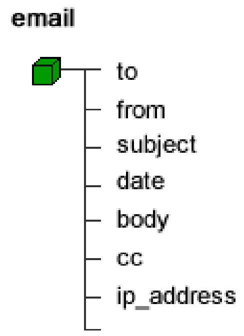

Figure 15 An object-type *email* described by a collection of fields.

Figure 16 A particular object, or node, contains *values* that match the predefined set of *keys* from Figure 15

4.4 Relationships

In Neo4j, the graph $G = (V, E)$, nodes $V = \{v_1, v_2, .., v_n\}$ and relationships E= $\{e_1, e_2, .., e_n\}$, a given relationship *rel* is given as an ordered pair of nodes (v_a, v_b), and denoted by a relationship type. For example, a small collection of types in Figure 17 are populated as in Figure 18. Examples of application-specific relationships may include **cover**(book, image) and **url**(book,url), as in Figure 19.

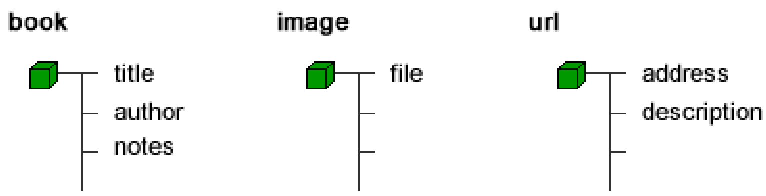

Figure 17 Example definitions for the node-types *book, image* and *url*.

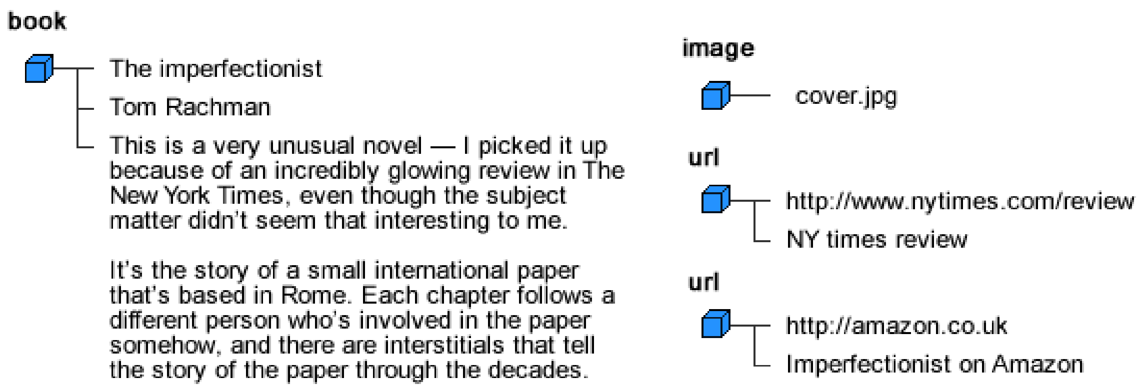

Figure 18 The types from Figure 17 populated with some sample data

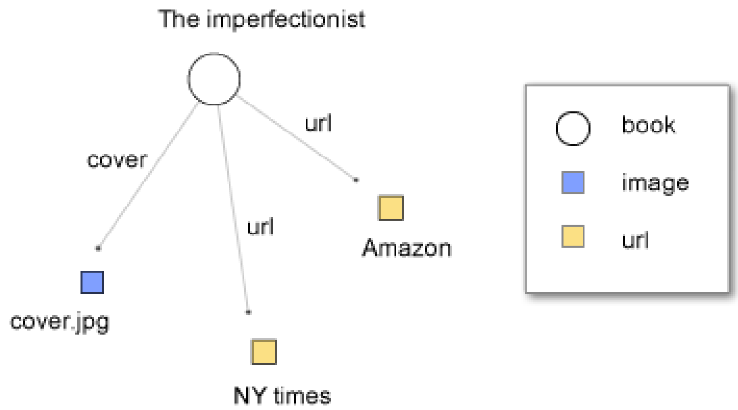

Figure 19 The data from Figure 18 forming a small graph. The relationships between (book, image) and (book, url) are denoted by a relationship type.

4.5 Data Representation

In a data-backed application we are free to produce *views* of the data, of a single data-point or as aggregations of data in a vicinity. A possible way of presenting the small graph from Figure 19 would be as structured content with a header, textual body, image and clickable links, observed in a browser would appear as Figure 20.

Figure 20 The data from Figure 19 viewed in a browser