

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems
(AMCIS)

8-2010

Collaboration and end-user information management tools

Mark Gregory

ESC Rennes School of Business, Mark.gregory@esc-rennes.fr

Follow this and additional works at: <http://aisel.aisnet.org/amcis2010>

Recommended Citation

Gregory, Mark, "Collaboration and end-user information management tools" (2010). *AMCIS 2010 Proceedings*. 586.
<http://aisel.aisnet.org/amcis2010/586>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Collaboration and end-user information management tools

Mark GREGORY

Department of Finance and Operations, ESC Rennes School of Business, 35065 Rennes, France
mark.gregory@esc-rennes.fr

ABSTRACT

Knowledge and information workers work as individuals within virtual team structures. Those teams may be small, or they may in fact form part of larger groups which continue to share similar objectives. Knowledge workers therefore acquire and manage personal and group information which they may choose to manage themselves using Personal and small-Group Information Management tools and techniques (PIM, GIM). This paper summarises some earlier research into PIM and GIM before addressing its *objectives*, which are to:

- Consider the use of PIM / GIM techniques as the basis for collaboration within those virtual teams
- Discuss collaboration and community involvement in the actual development of PIM/GIM tools
- Identify the need for formalisms and abstraction skills in effective personal information management.

Keywords (Required)

Personal information management (PIM); Group information management (GIM); situational application builders; end-user programming; abstraction; community development

PERSONAL AND GROUP INFORMATION MANAGEMENT: A BRIEF INTRODUCTION

As individuals and as team members, knowledge and information workers (Drucker 2000) acquire information, which they store in a number of more-or-less complex ways: some paper-based, but increasingly computer-based. See (Kelly 2006); (Teevan et al. 2006) for an introduction to personal information management (PIM).

Background

The author is engaged in a research programme into *knowledge and information management supporting people and groups* ("KIMSPAG"). Earlier papers have summarised work to date:

- (Gregory & Norbis 2008b) summarised then-current trends in the academic and practitioner's literature in the areas of knowledge representation and communication by individuals and small groups.
- (Gregory & Norbis 2008a) put forward a classification scheme for PIM tools based primarily on their data representation: e.g. spreadsheet, relational database and semantic web represented at the desktop level (Sauermaun et al. 2005)
- (Gregory & Norbis 2009a) described a personal auditing approach which has as its intention to assist users in improving their personal information management.
- (Gregory & Norbis 2009b) introduced situational applications as a way of building group information management systems.

This present paper summarises the earlier findings and then presents action research (Avison et al. 1999) carried out by the author and research colleagues in the context of the overall KIMSPAG research effort.

Information management for individuals and small groups

Different groups of knowledge workers (Drucker 2000) keep different kinds of personal information.

Many of us keep a wide range of personal data, which we classify or sub-divide into areas such as:

- Agenda: list of appointments
- Address book: our contacts
- To Do list

We also hold domain-specific data. Thus students and researchers keep more specialised (but still widely-used) data such as:

- Bibliography: reference list
- Reading notes
- Project logbook
- Individuals, teams and organisations need to carry out business processes; they have to Get Things Done: GTD - see (Allen 2003)
- To do this, they also need to Keep Found Things Found: KFTF – see (Jones 2007).

Work and the working context

The information stored by individuals and by small groups may be very straightforward in nature (that is, it may have simple syntax and semantics) or it may be very semantically rich. The richness of the data is then evidenced in complex structures with semi-formal semantics. Some tools provide their users with powerful querying (information retrieval) associated with compound filtering. This is comparatively manageable provided that a small number of users share a similar understanding of the data. The archetypal tool associated with this kind of information is the *spreadsheet*. The presentation is traditionally on the screen of the PC which hosts the spreadsheet, although Web based spreadsheets such as that in Google Docs have more recently become available (Herrick 2009).

When the group becomes larger, the volumes of data typically increase and the manageable semi-formal semantic richness falls. The semantics can be made more explicit if the data is constrained to follow certain rules. The archetypal tool associated with this kind of information is the *relational database*. Traditionally, relational databases have been either restricted in scope to a single PC or a very small number of locally-connected PCs - this is the requirement addressed by tools like Microsoft Office Access (Microsoft Office 2007); or a client-server architecture has been adopted to permit very large user populations to access a shared high-availability database.

The process of associating meaning with collections of information can be called *data modelling*, as in entity-relationship modelling (Date 2003). The process of designing processes to transform data into the information needed by its users is sometimes called *programming*, in cases where that processing is formalised into specific computer programs. That is not always necessary or it may not be cost-effective, in which case the processing may either be done by extracting the data into a spreadsheet where formulae can be applied to it; or the processing may be carried out manually by system users.

Migrating data to the Web: cloud computing, Web-Based Applications and personal information management

During the last two years, practitioners have become very focussed on the potential of what is sometimes called “*cloud computing*”, that is, the developing probability that personal, small-group and corporate data and the programs used to manage them will “migrate” from local client and server computers to reside in part or in whole on server computers accessed via the Web and the Internet. See for example (Miller 2009).

The significance of Cloud Computing and Web-Based Applications for personal information management

If information management increasingly resides in the cloud and simple web application builders become more usable, then the options for effective personal and small-group information management increase greatly.

Web-enabled, cloud-hosted and cloud-delivered end user development is likely to cause a considerable increase in the extent to which at least some knowledge workers can develop their own team-based information management.

PERSONAL AND SMALL GROUP INFORMATION AND KNOWLEDGE MANAGEMENT: SOME DEFINITIONS

Standard approaches to data and information management for individuals and small virtual teams

An earlier paper (Gregory & Norbis 2008b) suggested that individuals working in groups should be encouraged and educated to make better use of the available tools for information management and that the tools themselves should evolve into (or be replaced by) better ways of representing information and knowledge. That earlier paper suggested a classification scheme for tools based primarily on their data representation:

- It discussed *spreadsheets* and their shortcomings.
- It discussed *outliners*. (Outlining and outliners are discussed further below.)
- The paper also introduced the idea of what it called a *functional spreadsheet*, which simplifies and restricts the scope of spreadsheets to make them capable of formal representation. The idea was based on an insight documented in (Peyton Jones et al. 2003). Following (Nardi 1993), they identify the development of spreadsheets as end-user programming:

“For many people, the programming language of choice is a spreadsheet. This is especially true of people who are not employed as programmers, but write programs for their own use—often defined as “end-user” programmers. An end-user programmer is a teacher, an engineer, a physicist, a secretary, an accountant, a manager, in fact almost anything except a trained programmer. These people use computers to get their job done, but often they are not interested in programming *per se*.”

They go on to suggest that:

“*Spreadsheets lack the most fundamental mechanism that we use to control complexity: the ability to define re-usable abstractions.* In effect, they deny to end-user programmers the most powerful weapon in our armory.”

Our paper borrowed their idea of providing functions as first-class values, but simplified it in suggesting that a so-called “functional spreadsheet” approach be adopted in which rows are hierarchically organized in an outline that groups and sub-groups the data; and cells are limited to contain only values (such as text labels, dates and numbers) or the names of functions which may be applied either to all the values in a row; or to all the values in a group or sub-group.

- It discussed *relational databases*, in which all data is stored as relations or sets.
- It mentioned other approaches, including *object oriented databases*, *XML* documents, *RDF* and *OWL*. See (Davies et al. 2006).
- It introduced specific *PIM* (Personal Information Management) programs. Alternatively, there exist a number of personal information management tools, sometimes called PIM tools which can assist individuals and small groups in the storage and management of information. See (Kelly 2006) and (Teevan et al. 2006).

System building by and for non-experts

Another earlier paper (Gregory & Norbis 2008a) discussed in more detail the relative strengths of spreadsheets and databases and highlighted the “make or buy?” decision necessary when deciding on the best way to manage personal information.

Buying a standard PIM package was seen as straightforward but not always particularly satisfactory, as evidenced by the small number of computer users who make that investment (beyond the many who have Microsoft Outlook (Microsoft Office 2007) on their computer).

Optimizing personal data management was shown not to be simple, because of the necessity for structuring data in accordance with the constraints of the spreadsheet, database or user-definable structured PIM. A non-exhaustive list of available packages was analysed in terms of the kinds of basic functionality provided. The issue of ontological classification was introduced. Possible designs for improved PIM were discussed. The paper reached certain tentative conclusions:

- Data has little or no meaning except in context. Context gives meaning and removing context tends to remove meaning. People need to be encouraged to use tools which preserve context and thence meaning.
- Effective personal information management needs portable accessible computer resources that until very recently have not been portable enough. Notebooks have been heavy, expensive, dependant and lacking in autonomy. Smartphones were not all that smart!
- Computers can assist knowledge management. However, structuring knowledge is often alien to the way people want to work.
- No one single PIM approach will work for all groups of computer users. Some will prefer highly expressive, but more difficult to query and to manage, general solutions. Others will prefer very packaged, very restrictive approaches which dictate what kinds of information are stored. Many, perhaps most, will not be able to realise those benefits without knowledgeable “hand-holding”.

Earlier research questions

As result of this early work the following three initial research questions were identified for further investigation:

- Q1. The data-centred approach adopted by most PIMs is not necessarily well adapted to the working methods adopted by knowledge workers. Establishing what styles and functionalities appeal to (or repel) different types of users is not yet well understood.
- Q2. Current PIMs tend to emphasise one particular information management technique, to the exclusion of others. The absence of complementary information management techniques is one of the factors which cause knowledge workers to reject current PIMs.
- Q3. PIMs are not much used because PIMs either impose an ontology which does not correspond to the user's ontology, or do not permit that ontology to be made explicit and/or shared. The incorporation of explicit knowledge representation mechanisms which are tailored to their users' (plural) needs will make a PIM more useful: by beginning to turn it into a small-group knowledge manager.

PERSONAL AND SMALL GROUP INFORMATION AND KNOWLEDGE MANAGEMENT: CHALLENGES

Why information management should be personal

Businesses invest in information systems used by a part (small or large) of their communities of internal users and often by other stakeholders (notably, their clients and supply-chain partners). Shared systems, whether business or infotainment, are almost all used in whole or in part by individuals whose usage is personal and where option settings and data are personal.

We teach students who are learning for the first time about business information systems, the importance of the Three U's, that is, that Usefulness and Usability give rise to actual Use. This is effectively a reformulation and application of Davis's technology adoption model (Davis 1989) to information systems; Davis uses the terms perceived usefulness, perceived ease of use, and user acceptance.

Why information management is only rarely purely personal

Comparatively little information stored and intermediated by Information and Communications Technology (ICT) is purely personal in its use and usefulness. Even the apparently intrinsically personal is often in fact an instance only of shared information: telephone numbers, email addresses, diary appointments, usually involve sharing or copying information "personal to" (or "owned by") other people. A copy of certain information may indeed be purely personal – but it often loses utility precisely because it is a personal copy (and is therefore not always in step with the "original" data).

It has long been accepted that data duplication should, where possible, be avoided – this was a major motivation in the widespread adoption of database management systems a generation ago; see for example (Date 2003). The very recent explosion of interest in and dependence on 'cloud computing' is accompanied by an increasing dependence on cloud-hosted information systems which offer shared access to services such as contact management – e.g. (Plaxo 2010). Cloud computing itself is as yet documented more in popular sources such as (Miller 2009); (Fingar 2009); (Sapir 2009) than in academic literature.

ICT-based business information management and its inherent asymmetry: the role of the user and the role of the expert

There is extensive literature on end-user computing and indeed on end-user programming; see for example (Regan & O'Connor 1994). Effective end-user computing is frequently supported within organisations by extensive ICT and human infrastructure (e.g. help desks). There is a strong desirability of; and / or pragmatic necessity for; minimising the need for expert intervention. It is ultimately the user who benefits from taking a high degree of responsibility for, and appropriate authority in using, a system. However, a systems expert still needs to be available to carry out specialist tasks.

The "make or buy" quandary

The ways in which a company can procure the business applications necessary for its operations include combinations of three basic approaches which are already widely accepted, and a fourth:

1. *Bespoke* (custom) development; this requires technical skills
2. Purchase and use of *packaged* applications

3. **Systems integration** – composing applications from different components found on the web; this requires technical skills
4. **So-called “end user programming”**, (Nardi 1993); (Regan & O'Connor 1994) **which can be done by business professionals.**

It is often possible to procure a packaged solution. Thus in the preparation of this paper, (Zotero 2010) was used to manage citations and references. This replaces an earlier approach based on Excel (Microsoft Office 2007). However, packages necessarily restrict the generality of approach which might be desirable or necessary.

There are many Information Systems requirements which are on a relatively small scale, and where reasonably-courageous individuals, sometimes referred to as “power users”, set out to build their own small-scale Information Systems. Regan has written extensively on this phenomenon, sometimes known as “*end-user programming*”; see for example (Regan & O'Connor 1994).

Situated applications were first identified by (Shirky 2004). (Cherbakov et al. 2007) describe what are now known as situational applications (SAs) in these terms:

“The recent rise of grassroots computing among both professional programmers and knowledge workers has highlighted a development approach in which those who best understand the business problem at hand develop rapid solutions without the overhead and formality of traditional IT methods. The new breed of situational applications (SAs), often developed by amateur programmers in an iterative and collaborative way, shortens the traditional edit-compile-test-run development life cycle. SAs have the potential to solve immediate business challenges in a cost-effective way, capturing the part of IT that directly impacts end users and addressing the areas that were previously unaffordable or of lower priority.

”

They go on to suggest that SAs can be characterised as:

- Developed to address the situation at hand
- Often built by non-traditional, casual programmers with little up-front emphasis on reliability, scalability, maintainability, and availability
- Developed in short, iterative cycles measured in days or weeks rather than months or years, focusing on time-to-value
- Usually information-centric
- Built as so-called “mashups” (Gruber 2007) or as Web-based database applications

We can summarise the positioning of situational applications in a mapping the author suggested in (Gregory & Norbis 2008b) for the “applications space”:

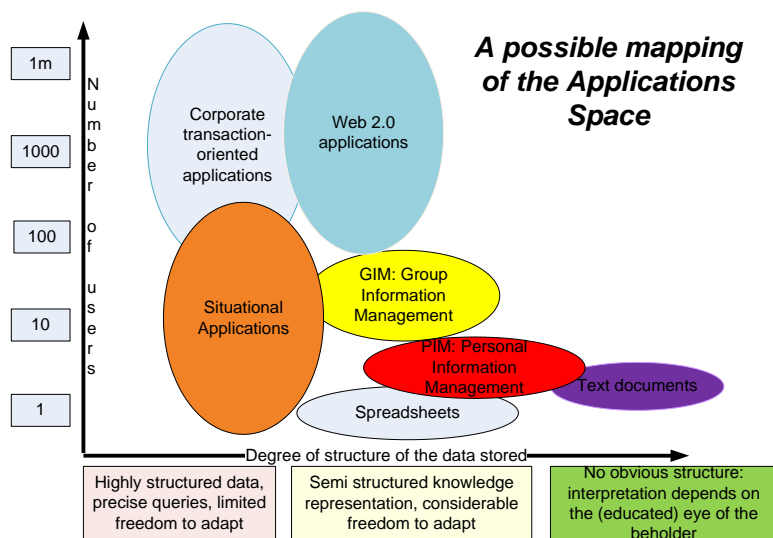


Figure 1: A possible mapping of the applications space

Social networks and collaborative development

There has been an explosion of use of so-called “social networks”. (Hoadley et al. 2010) report how “Increasingly, millions of people, especially youth, post personal information in online social networks (OSNs)”. They blog, tag, establish folksonomies and carry out collaborative development of wiki-based documentary resources (Gruber 2007).

Why is it so difficult to build end-user oriented systems?

System building is a process in which someone – be she professional developer or end-user – constructs a concrete representation of a desired target system which “sufficiently” corresponds to the behaviour required of it in the real-world system of which it will become an embedded part. We can in general recognise two phases in this process: an analysis of requirements – that is, modelling what is required of the target system; then a synthesis process in which components are acquired or created, then assembled and tested. Both the analytical and the synthesising phases require a capacity for abstraction which is variably present in generally well-educated adults.

(Kramer 2007) states that:

“Formal modeling and analysis is a powerful means for practicing abstract thinking and consolidating students’ ability to apply abstraction. Modeling is the most important engineering technique; models help us to understand and analyze large and complex problems. Since models are a simplification of reality intended to promote understanding and reasoning, students must exercise all their abstraction skills to construct models that are fit for purpose. They must also be capable of mapping between reality and the abstraction, so as to appreciate the limitations of the abstraction and to interpret the implications of model analysis.”

He reports that he has been unable to find any existing appropriate tests for measuring students’ abstract thinking and abstraction skills. He also suggests that:

“Given a model, students find it very helpful in clarifying the important aspects of the problem and in using a model checking tool to reason about its properties and behavior. However, some still seem to find it extremely difficult to construct the models themselves.”

And here we might be at the heart of the necessity for collaboration: a knowledge diffusion or learning process in which expertise is transferred, or learning facilitated, in a partnership between someone already skilled in information management and someone who needs to learn and apply information management skills in their particular context.

AN EXAMPLE OF DIRECT KNOWLEDGE TRANSFER FROM DEVELOPER TO END-USER PROGRAMMER: QRIMP USED TO BUILD LODE

So people need people: for knowledge transfer, effective learning and mutual encouragement. This may take the form of conventional teaching and learning, or of peer-to-peer informal training and assistance.

As part of research into personal and group information management, but also as a practical tool that was needed in his teaching situation, the author built LODE. LODE (Learning Outcome Demonstration Environment or Learning Object Deployment Environment) is a situational application built using a situational application builder.

The situational application builder used to build LODE is called Qrimp (Qrimp 2010). Qrimp can be viewed as a cloud-based relational database (and associated tools) by means of which motivated and guided end users can create effective web-accessible database systems.

In its first incarnation, LODE was used by over 400 students as a secure online system for the submission of homework and project work which was subsequently assessed by teachers. LODE has already taught the following lessons:

- It is possible to design, build and deploy web-accessible information-centric applications in a matter of weeks and with a very high degree of reliability and availability.
- We as teachers needed the active transitional support of a Qrimp principal.
- It is easy to evolve the application because, being information-centred and cloud-based, deployment issues are minimal.

LODE has thus demonstrated the need for a partnership between well-educated end users on the one side and a skilled developer on the other.

AN EXAMPLE OF COMMUNITY KNOWLEDGE TRANSFER: INFOQUBE

An aspect of our research outlined earlier was summarised as: “Some will prefer highly expressive, but more difficult to query and to manage, general solutions... Many, perhaps most, will not be able to realise those benefits

without knowledgeable 'hand-holding'". The author has himself made extensive use of a product in development called InfoQube (InfoQube 2010), and observed its development and the role played by its community of users in that development.

Background: Outlining and Outliners

Outlining is a long-established approach to structuring and writing text (Price 1999).

An *outline* is a hierarchical way to display related items of text to graphically depict their relationships. *Outlining* is a technique which may be implemented in general office programs or in specific computer programs known as "*outliners*". An *outliner* is a special text editor that allows text to be structured as an outline. Outliners are typically used for computer programming, collecting or organizing ideas. Outlining is the technique widely used in programs such as Microsoft Office PowerPoint, in which the main headings of a presentation appear as separate slides and on each slide appear points and sub-points; and in Word's Outline view. See (Microsoft Office 2007).

In outlining, a data item is given meaning by being shown in its owning hierarchy. Thus a person's surname is a component of a composite Contact object.

Realised in Word and formatted in a particular way, an outline has an appearance similar to:

11.	Semantic Web: Web 3.0?
11.1.	What is the Semantic Web?
11.2.	Purpose
11.3.	Realisation: going beyond the hypertext web
11.3.1.	Markup
11.3.2.	Descriptive, and extensible
11.3.3.	XML, XML Schema, RDF, OWL, SPARQL: the W3C
	Semantic Web Layer Cake
11.3.4.	Enhancing the usability and usefulness of the Web and its
	interconnected resources through:
11.3.5.	The Issue: whose ontology?
11.4.	Semantic Web: current state of the art
11.4.1.	The Simile project (MIT)
11.5.	Semantic desktop: the semantic Web represented at the small-group level
11.5.1.	Introduction
11.5.2.	General description
11.5.3.	Different interpretations of the semantic desktop
11.5.4.	Relationship with the Semantic Web
11.6.	"Web 2.0" and Social Networking
11.7.	Semantic wikis
11.7.1.	Semantic Wiki Interest Group

Figure 2: Outline formatted as a hierarchy of points, sub-points, sub-sub-points.

Here, the owner in the hierarchy as shown is 11. Semantic Web. It is the eleventh point in a document – it is implicitly owned by the document of which it forms a part.

It owns items 11.1, 11.2, 11.3, ...

11.3 owns 11.3.1, 11.3.2, ...

The owning item for 11.2, 11.3 ... is 11.

The relative positioning of an item conveys meaning in that the label of the owner classifies or otherwise gives contextual information concerning the owned item.

Outliner programs go further; thus (Ecco 1997) permits the definition of forms to impose some order on anarchy. Further, a data item can participate in more than one hierarchy; an appointment can appear in an overall agenda or calendar, but also be linked to the name of each participant in the meeting. Effectively, the same datum is classified in more than one way.

InfoQube

InfoQube (InfoQube 2010), a Microsoft Windows application developed by NeoTech Systems, is a one-pane outliner (in that hierarchy and rich text can be displayed in the outline itself). It is also technically a two-pane outliner, in that any item in an outline can also be associated with an arbitrary amount of fully-formatted HTML. Standard features include task and project management, a calendar which can display any item having a date attribute, basic concept

mapping and crucially a *grid of values associated with any given item*. The definition of the grid, and therefore the associated semantics, are entirely under user control. Grids display item values as a row; forms display them as a column. Technically, a grid is a view of underlying system-maintained relational tables based on nested SQL queries. Thus any field definition (and associated values) can occur multiply and simultaneously in any of an arbitrary number of grids. A grid can also be linked to dynamically from Word or Excel.

IQ implements row-level and column-level equations. The syntax for these is based on Visual Basic (the default scripting engine is VBScript). Further, the program provides a repertoire of system-defined functions. Users with programming skills can program their own functions with the built-in VBScript editor. Thus IQ can be regarded as a functional spreadsheet in the sense identified above. Information presentation facilities include sorting, multi-criteria filtering, summary tables, charts, Gantt charts, pivot tables and conditional formatting. Web clippings and emails can be incorporated within the HTML pane. Windows file hierarchies can be linked to dynamically.

IQ was originally developed to meet a specific company's needs. It takes its inspiration from (Ecco 1997), whose development ceased in 1997 but which retains many enthusiastic users. Technically, IQ is built using Visual Basic on a packaged Microsoft Office Access database (Microsoft Office 2007).

IQ has yet (April 2010) to achieve a formal release. Instead, a growing community of largely enthusiastic users acts as a bank of beta testers. Minor releases occur about fortnightly; major dot releases once every two months or so. The incidence of minor bugs is low and major bugs (ones causing data loss) are almost unheard of. As an Access database application, IQ is inherently multi-user and supports some concurrency.

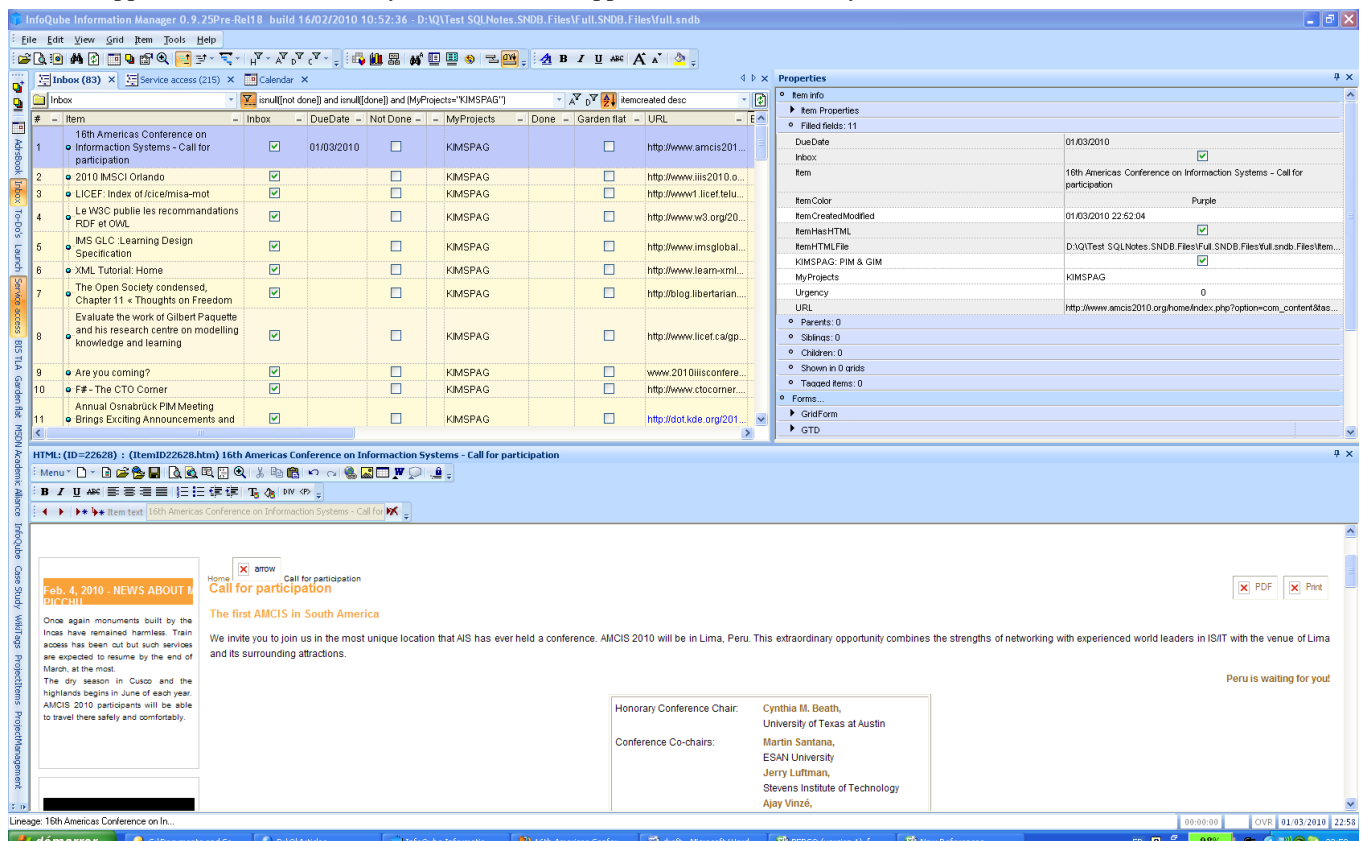


Figure 3: InfoQube screen shot

- The software can quickly be modified as users make suggestions; provided that those requests fall within the "global" architectural vision of its developer (this approach retains architectural coherence and protection of data investment). Both developers and users benefit from this close collaboration.
- The user interface is very flexible and supports scripting. This allows its use as a framework for custom software solutions (similar to Microsoft Access (Microsoft Office 2007)).
- The underlying database engine is inherently multi-user and supports replication, so the collaboration applies not only to the development of the product but also to the actual

finished product use in supporting both connected (i.e. LAN-based) and disconnected (i.e. Web) topologies.

- InfoQube permits its users to clip content from browsers and from email clients. Thus an item is stored which retains a hyperlink to its original source; furthermore the content of the original webpage or email can be stored in the HTML pane referred to above. This content can help the human user to preserve context and thus retain more meaning.
- **Exchange between developer and IQ users, and between IQ users**

An infrastructure was put in place to connect the users with the developers. NeoTech Systems maintains a forum using the Drupal content management system (Drupal 2010). A small but very active community of users is associated with the site. New users are helped to get started by others with longer experience. NeoTech is very responsive to user suggestions for new functionality provided they are architecturally coherent. Being in beta, the community can and does have a large impact on the development

EVALUATION AND QUESTIONS FOR FURTHER RESEARCH

We have suggested need for a partnership between well-educated end users on the one side, and a helpful community or skilled developer on the other, and shown how this can be beneficial in specific cases.

The work presented in this paper impacts the research questions earlier identified in the following ways:

1. *The data-centred approach adopted by most PIMs is not necessarily well adapted to the working methods adopted by knowledge workers. Establishing what styles and functionalities appeal to (or repel) different types of users is not yet well understood.*

We now have a better understanding of the reactions of student users to a GIM built using Qrimp. However, in-depth ethnographic study of student reactions, and action research as we enhance the application, is now required - and this is the subject of ongoing work by the author.

2. *Current PIMs tend to emphasise one particular information management technique, to the exclusion of others. The absence of complementary information management techniques is one of the factors which cause knowledge workers to reject current PIMs.*

We have identified at least one PIM (InfoQube) which offers a semi-relational database style, outlining, inter-item linking, unification of the system file store and the meta-data it contains about those file and web links, functional-spreadsheet-like capability, classification and wiki-style tagging, and pivot tables which can be dynamically linked to actual spreadsheets. We have indicated the need for peer-to-peer mutual assistance when seeking to maximise the benefits offered by such functionality. Thus this PIM can be said to meet the preference suggested above for highly expressive, but more difficult to query and to manage, general solutions. Forum users (and importantly, former forum users) report a steep learning curve. Again, in-depth ethnographic study is now required and this is the subject of ongoing work by the author.

3. *PIMs are not much used because PIMs either impose an ontology which does not correspond to the user's ontology, or do not permit that ontology to be made explicit and/or shared. The incorporation of explicit knowledge representation mechanisms which are tailored to their users' (plural) needs will make a PIM more useful: by beginning to turn it into a small-group knowledge manager.*

Although it is possible and indeed straightforward to build tagging and classification schemes using both Qrimp and IQ, we have not yet done so in a demonstrable form. This again is the subject of ongoing work by the author.

ACKNOWLEDGEMENTS

The author acknowledges the considerable help provided by Randall Minter as Chief Executive and developer at Qrimp, Inc. and their generous provision of computing resources. The author also acknowledges the great help and

encouragement provided by Pierre-Paul Landry as principal at NeoTech Systems; and of InfoQube forum users, especially Menez Chapleau.

REFERENCES

- Allen, D., 2003. *Getting things done: The art of stress-free productivity*, Penguin Books.
- Avison, D.E. et al., 1999. Action research. *Communications of the ACM*, 42(1), 97.
- Cherbakov, L. et al., 2007. Changing the corporate IT development model: Tapping the power of grassroots computing. *IBM Systems Journal*, 46(4), 743.
- Date, C., 2003. *An Introduction to Database Systems* 8th ed., Addison Wesley.
- Davies, J., Studer, R. & Warren, P., 2006. *Semantic web technologies*, John Wiley & Sons.
- Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340.
- Drucker, P.F., 2000. Knowledge-worker productivity: The biggest challenge. *The knowledge management yearbook 2000–2001*.
- Drupal, 2010. Drupal: drupal.org. Available at: <http://drupal.org/> [Accessed April 26, 2010].
- Ecco, 1997. ecco_pro : Ecco Pro Software :: support & extension. Available at: http://tech.groups.yahoo.com/group/ecco_pro/ [Accessed April 26, 2010].
- Fingar, P., 2009. *Dot.Cloud*, Meghan-Kiffer Press.
- Gregory, M. & Norbis, M., 2009a. Auditing Personal Information Management. In UKAIS 2009. St. Anne's College, Oxford.
- Gregory, M. & Norbis, M., 2009b. Evaluating Situational Applications Builders. In CITSA 2009: The 6th International Conference on Cybernetics and Information Technologies, Systems and Applications. Orlando, FA.
- Gregory, M. & Norbis, M., 2008a. The business of personal knowledge. In 8th International Conference on Knowledge, Culture and Change in Organisations. Cambridge University, United Kingdom.
- Gregory, M. & Norbis, M., 2008b. Towards a Systematic Evaluation of Personal and Small Group Information and Knowledge Management. In 5th International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2008. Orlando, FA.
- Gruber, T., 2007. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web and Information Systems*, 3(1), 1–11.
- Herrick, D.R., 2009. Google this!: using Google apps for collaboration and productivity. In

- Proceedings of the ACM SIGUCCS fall conference on User services conference*. pp. 55–64.
- Hoadley, C.M. et al., 2010. Privacy as information access and illusory control: The case of the Facebook News Feed privacy outcry. *Electronic Commerce Research and Applications*, 9(1), 50–60.
- InfoQube, 2010. InfoQube Information Management System. Available at: <http://www.infoqube.biz/> [Accessed April 26, 2010].
- Jones, W.P., 2007. *Keeping found things found: The study and practice of personal information management*, Morgan Kaufmann Pub.
- Kelly, D., 2006. Evaluating personal information management behaviors and tools. *Communications of the ACM*, 49(1), 86.
- Kramer, J., 2007. Is abstraction the key to computing? *Communications of the ACM*, 50(4), 42.
- Microsoft Office, 2007. Office Online Home Page - Microsoft Office Online. Available at: <http://office.microsoft.com/en-gb/default.aspx> [Accessed April 26, 2010].
- Miller, M., 2009. *Cloud computing : Web-based applications that change the way you work and collaborate online.*, Indianapolis, IN: Que, 2009. Available at: <http://www.worldcat.org/title/cloud-computing-web-based-applications-that-change-the-way-you-work-and-collaborate-online/oclc/474531630> [Accessed April 26, 2010].
- Nardi, B.A., 1993. *A small matter of programming*, MIT Press.
- Peyton Jones, S., Blackwell, A. & Burnett, M., 2003. A user-centred approach to functions in Excel. In *Proceedings of the eighth ACM SIGPLAN international conference on Functional programming*. p. 176.
- Plaxo, 2010. Plaxo - votre carnet d'adresses pour la vie. Available at: <http://www.plaxo.com/> [Accessed April 26, 2010].
- Price, J., 1999. *Outlining goes electronic*, Greenwood Publishing Group.
- Qrimp, 2010. Qrimp - Advanced Web Database. Available at: <http://www.qrimp.com/> [Accessed April 26, 2010].
- Regan, E.A. & O'Connor, B.N., 1994. *End-user information systems*, Macmillan Pub. Co.
- Sapir, J., 2009. *Power in the Cloud*, Meghan Kiffer Pr.
- Sauermann, L., Bernardi, A. & Dengel, A., 2005. Overview and outlook on the semantic desktop. In *Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference*. pp. 1–19.

Shirky, C., 2004. Situated software. *Firstly published on the Networks, Economics, and Culture mailing list on March, 30, 2004.* Available at: http://www.shirky.com/writings/situated_software.html [Accessed April 26, 2010].

Teevan, J., Jones, W. & Bederson, B.B., 2006. Personal information management. *Communications of the ACM*, 49(1), 40–43.

Zotero, 2010. Zotero 2.0 Documentation. Available at: <http://www.zotero.org/support/2.0> [Accessed April 23, 2010].