Semantic Web Approach to Personal Information Management on Mobile Devices

Ora Lassila
Nokia Research Center
Cambridge, Massachusetts, USA

I. Introduction

The role and nature of *Personal Information Management* (PIM) have changed in recent years. Where the "original" PIM mostly encompassed managing the user's calendar, address book and to-do list, and where most (if not all) of the data was produced by the user herself, the "modern" PIM encompasses other kinds of data as well: RSS feeds, Web bookmarks, various media files and their associated metadata, etc. There is also a lot more data to manage and manipulate, and the bulk of it typically comes from other sources or is otherwise fragmented across multiple systems and devices [7]. PIM and *social networking* are also intimately linked as users increasingly desire to be "connected" to their family, friends and colleagues [5].

The new challenges to PIM software and systems are many: New types of information must be handled, and there are new types of devices available to users – suggesting new types of user interfaces.

We observe that PIM, fundamentally, is about *information*, and any advances in the representation, storage, manipulation and presentation of information may benefit PIM. In particular, in this paper, we will adopt the *Semantic Web* [2] as our representational basis, and present some results of applying this approach in the context of mobile devices. Underlying the discussion is a desire to ask (and answer) some "deep" questions about our use of information technology.

II. About the Semantic Web

The Semantic Web is often billed as the "next generation of the World Wide Web", but this definition may be too narrow. The original vision for the Semantic Web [2] is really a broad vision of the future of *personal computing*, very much centered around the use of computing in ordinary people's everyday lives. Thus, taking Semantic Web technologies and applying them to PIM is a natural progression not only in our quest to understand the applicability of these semantic technologies, but also in the evolution of PIM towards comprehensive management of (and access to) *all* the information that will help people manage their lives.

The subtext here is that the Semantic Web technologies, as “advertised” in the original vision, are an enabler for software systems that do more *on behalf of* their human users, and in our ever more complicated and complex lives, this – we believe – is the desired direction of development of *Information and Communication Technologies* (ICT) in general: Until now, our use of ICT has been very much in line with how humans have used *any* technologies since prehistoric times – as *tools*; the next fundamental step in the evolution of our use of technology should be to break this limitation.[1] To actually be able to delegate tasks to an automated system requires a level of detail and fidelity in the description of the world (and the tasks) that may not be practical; later, we will argue that new mobile devices can be well suited to the *automatic* acquisition of some of the *contextual* information necessary to make proficient task delegation a reality.

At a more practical and concrete level, Semantic Web technologies offer several benefits to building information-centered and information-intensive applications:

1) **Uniformity of data:** A *uniform data model*[2] makes it easier to exchange information and to integrate applications. The traditional model of *software applications* is really a software engineering vehicle for bundling functionality. Applications also serve as the encapsulations (and embodiments) of *data semantics*: In order to manipulate certain kind of data you need the specific application that embodies the semantics of this data. Semantic Web technologies allow us to start separating applications from data semantics, and associating the semantics with the data itself. We also *expose* the semantics for external scrutiny, something that is not possible with the “black box” procedural semantics of contemporary applications.
2) **Future-proofing:** Semantic Web technologies are helpful when building systems capable of dealing with *unanticipated data* – that is, information (schemata) not considered when these systems were designed.
3) **Data integration:** Semantic Web technologies simplify the task of *integrating information from multiple sources*, providing a framework where the actual integration details of “mash-ups” can be worked out automatically (rather than by a programmer). Furthermore, because of the simple metamodel of RDF [15], the basic building block of Semantic Web data, tracking *data provenance* becomes practical, potentially making “mash-ups” more useful since one can still identify the sources of various bits of information at a very fine granular level. This also makes it possible to *explain* system behavior and outcomes, a characteristic laid out in the original Semantic Web vision.

While offering new benefits, Semantic Web technologies also present us with considerable challenges [12], including:

1) **Cultural resistance:** Reluctance of developers to adopt these technologies for a multitude of reasons largely charac-

[1] Stating this inevitably leads to a discussion about the role of *Artificial Intelligence* (AI) with the Semantic Web. In this paper we do not intend to get deep into this discussion.

[2] We mean a uniform *metamodel*, since the “let all flowers bloom” design principle of the Semantic Web with respect to schemata and ontologies makes the data itself anything but uniform. We consider this to be a Good Thing.

terizable as “cultural” or “religious”. This is quite similar to the phenomenon known as the “Artificial Intelligence Winter” [16, for example].

2) **Lack of business models:** Many of the benefits of Semantic Web technologies are *indirect* and do not lend themselves well to direct generation of revenue [18].
3) **Difficult programming models:** It has not been clear exactly how developers should exploit Semantic Web technologies and how applications that leverage the Semantic Web’s strength and flexibility should be written [12].[3]

In this paper we assume that we can overcome the first two challenges (or at least mitigate them); we will discuss some approaches to dealing with the third (the reader is referred to [12] for a more thorough analysis).

There are already results in applying Semantic Web technologies to PIM, including social networking [4], [3] and so-called “Semantic Desktops” [19], [5], [14]. In this paper, we discuss the opportunities and challenges brought by applying Semantic Web in the context of small, handheld devices.

III. About Mobile Computing

The recent emergence of handheld devices as powerful computational platforms (e.g., “smartphones”) presents an opportunity to build sophisticated PIM applications that can take advantage of some the unique characteristics of these devices:

1) **Always with you, always “on”, always connected:** Conceivably we could use these handheld devices for many things that our traditional desktop (or even laptop) computers are simply not suited for (due to their physical size, their lack of ubiquitous connectivity, etc.). For example, taking quick notes (say, writing down a phone number or an address) is something many people have used pen and paper for because it simply is too cumbersome to bring out the laptop computer – more specifically, the “investment” is too great in comparison to the perceived benefit. A small device that is always with you and always “on” is better suited for this task.[4]
2) **Location awareness:** A device that moves and is aware of its (geographical) location can offer this information to the benefit of software applications. Certain actions only make sense in specific locations (or can be customized based on the location). In technical terms, in an open-ended world the location provides a way of limiting possibilities (say, limiting the search for specific types of physical services) to the point that certain applications become technically possible and realistic. *Service Discovery* is a good example: Traditionally discovery takes advantage of network topology, but in a world with a multitude of physical services this may simply not be enough.
3) **Context awareness:** The notion of *context* [6], [13] involves “understanding” aspects of the situation that describe the user’s current environment, task, goals and actions. Location is a limited special case of context; modern smart-

[3]A corollary to this is that once we start employing complex representations of information, also the demands on *user interaction* are elevated.

[4]The caveat is that this statement may be justifiable only if the UI *equals or exceeds* that of pen and paper.

phones, though, offer various "sensors" and other mechanism for determining context beyond mere location.[5] Similarly to location, context can be used to limit possibilities and direct search (of functionality, services, etc.).

With the added capabilities of handheld devices, and the proliferation of various computational devices in general, we are starting to see the deployment of *Ubiquitous Computing* [25]. Here Semantic Web technologies have also been found beneficial in addressing some of the inherent tough problems, such as service discovery, composition and adaptation [1], [17] as well as various interoperability issues in general [8].

While providing some new benefits to application developers, contemporary handheld devices are still lagging behind their desk-bound counterparts in terms of CPU performance, memory capacity and – in some cases – network bandwidth and latency. The devices are also physically more limited with much smaller displays and restricted keyboards. Over time, we believe, most of these limitations can be overcome. The real limitation, however, is that the use of these devices often happens in situations where the user is "attention-constrained", engaged in some other potentially demanding and/or distracting activity (such as driving a car); this places considerable demands on (new types of) user interaction [9].

IV. Some Use Cases

In pursuit of better PIM support, we have experimented with Semantic Web technologies and addressed many of the issues outlined above. Some of these experiments are described below.

A. General Data Browsing

The *OINK* system [10], [11] is a general data browser that can be used as a platform for building applications where the user interacts with rich data; any data (type) can be provided with customized way of visualization, with the assumption that when such customization is not found, the system uses a generic visualization that tries to make use of any known data schemata to provide a human-readabe result (see Fig. 1). In this respect, the system exhibits "best effort" behavior in data visualization.

The OINK system is also capable of performing data management and automatic data integration based on the use of a reasoning engine[6] and a path-based query engine [12]. To improve the ways in which data is accessed and presented, the system is "policy-aware", allowing all data to be controlled by a set of (arbitrarily complex) context-adjusted policies [21], [22].

B. Supporting User Input

The *Jourknow* system provides support for "lightweight" note-taking, based on the use of written notes and simple, "pseudo-natural" language grammars [23]. The basic idea is that we interpret the user's notes to create structured data (represented in RDF), with the expectation that this structure and its associated semantics allow the notes to be acted on. In addition, the system makes use of a continuously running "activity capture"

[5] http://research.nokia.com/projects/activity_monitor is a good example.

[6] Our reasoner implements the semantics for "RDF++", an extension of RDF(S) [12].

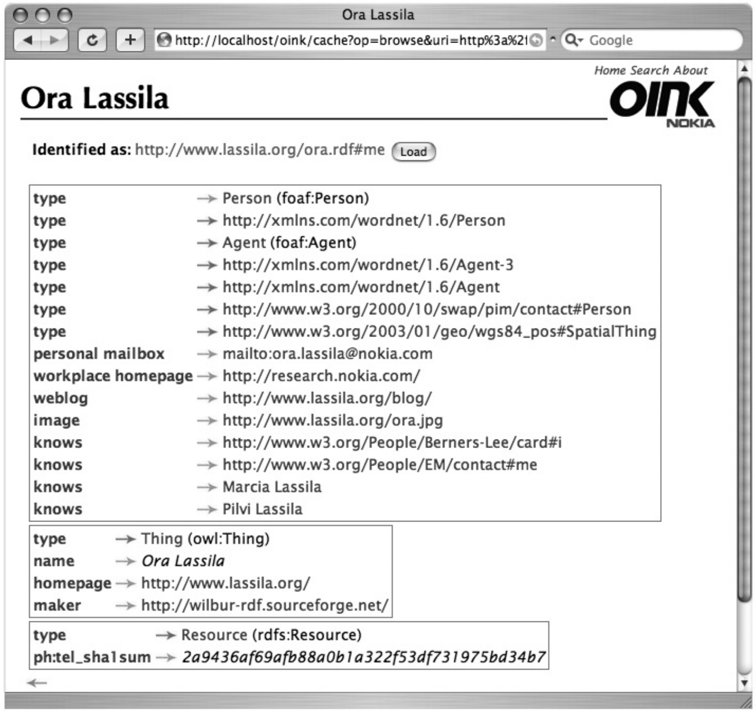

Fig. 1. A typical data view in OINK, shown in a regular Web browser

engine that attempts to identify the situational contexts in which the notes were taken [24]; these contextual "cues" can later be used to enhance the user interface for searching through notes. Fig. 2 shows an example of Jourknow running on a Nokia S60 phone.

C. Conversational User Interfaces

As a departure from graphical, direct-manipulation user interfaces we are investigating conversational, *dialogue-based* user interaction. The dialogues, conducted between the user and the system in spoken, natural language, are generated from the rich (PIM) information represented using RDF. This work is based on our earlier work on conversational user interfaces [20]; the ultimate goal is to provide "dialogue-management" that can span multiple subject domains, the same way conversations between people are often conducted.

V. Conclusions

Personal Information Management, with its new, expanded scope, is now almost synonymous with "personal computing". Not only does PIM now cover a broader spectrum of data (some directly personal, some indirectly via social connections, etc.), but also the variety of devices on (and through) which the users manipulate their personal data has grown, and most users have *several* devices that they use regularly. This introduces great challenges to how we build PIM systems,

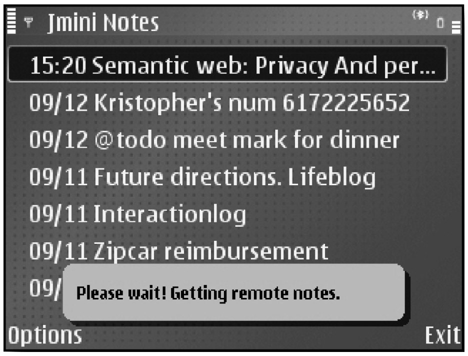

Fig. 2. Jourknow on an S60 phone

starting from various data management and representation issues and spanning through to user interaction problems. Mobile devices are becoming ubiquitous, and with their increased performance and various ways of sensing their environment, they are better and better suited to building increasingly capable PIM systems. Naturally, the ultimate goal is not PIM *per se*, but rather to realize technology that can help users by simplifying their everyday lives.

References

[1] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. McDermott, D. Martin, S. A. McIllraith, S. Narayanan, M. Paolucci, T. Payne, and K. Sycara. DAML-S: Web Service Description for the Semantic Web. In I. Horrocks and J. Hendler, editors, *The Semantic Web - ISWC 2002, 1st International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 348–363. Springer Verlag, 2002.

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

[3] J. G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards Semantically Interlinked Online Communities. In *Proc. 2nd European Semantic Web Conf.*, number 3532 in Lecture Notes in Computer Science, pages 500–514, Heraklion, Greece, 2005. Springer-Verlag.

[4] D. Brickley and L. Miller. FOAF Vocabulary Specification. http://xmlns.com/foaf/0.1/, Sept. 2004.

[5] S. Decker and M. Frank. The Social Semantic Desktop. Technical Report DERI-TR-2004-05-02, DERI, 2004.

[6] A. Dey, G. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16:97–166, 2001.

[7] D. R. Karger and W. Jones. Data unification in personal information management. *Commun. ACM*, 49(1):77–82, 2006.

[8] O. Lassila. Serendipitous Interoperability. In Eero Hyvönen, editor, *The Semantic Web Kick-off in Finland – Vision, Technologies, Research, and Applications*, HIIT Publications 2002-001. University of Helsinki, 2002.

[9] O. Lassila. Applying Semantic Web in Mobile and Ubiquitous Computing: Will Policy-Awareness Help? In L. Kagal, T. Finin, and J. Hendler, editors, *Proceedings of the Semantic Web Policy Workshop, 4th International Semantic Web Conference*, pages 6–11, Galway, Ireland, Nov. 2005.

[10] O. Lassila. Browsing the Semantic Web. In *17th International Conference on Database and Expert Systems Applications (DEXA'06)*, pages 365–369, Krakow, Poland, 2006. IEEE Computer Society.

[11] O. Lassila. Generating Rewrite Rules by Browsing RDF Data. In *Proceedings of the Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML 2006)*. IEEE Computer Society, 2006.

[12] O. Lassila. *Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data*. PhD thesis, Helsinki University of Technology, November 2007.

[13] O. Lassila and D. Khushraj. Contextualizing Applications via Semantic Middleware. In *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*. IEEE Computer Society, 2005.

[14] O. Lassila, D. Khushraj, and R. R. Swick. Spontaneous Collaboration via Browsing of Semantic Data on Mobile Devices. In S. Decker, J. Park, L. Sauermann, S. Auer, and S. Handschuh, editors, *Proceedings of the Semantic Desktop and Social Semantic Collaboration Workshop (SemDesk 2006)*, number 202 in CEUR Workshop Proceedings, Athens, GA, 2006. CEUR-WS.org.

[15] O. Lassila and R. R. Swick. Resource Description

Framework (RDF) Model and Syntax Specification. W3C Recommendation, World Wide Web Consortium, Feb. 1999.

[16] J. Markoff. Behind Artificial Intelligence, a Squadron of Bright Real People. *New York Times*, October 14th, 2005.

[17] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. A. McIllraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. OWL-S: Semantic Markup for Web Services. W3C Member Submission, World Wide Web Consortium, 2004.

[18] D. Provost. Hurdles in the Business Case for the Semantic Web. Master's thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, June 2004.

[19] L. Sauermann. The Gnowsis: Using Semantic Web Technologies to Build a Semantic Desktop. Master's thesis, Technische Universität Wien, 2003.

[20] S. Seneff, M. Adler, J. R. Glass, B. Sherry, T. J. Hazen, C. Wang, and T. Wu. Exploiting Context Information in Spoken Dialogue Interaction with Mobile Devices. In *Proc. Intl. Workshop on Improved Mobile User Experience (IMUx 2007)*, Toronto, Canada, 2007.

[21] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. A Semantic Context-Aware Access Control Framework for Securing Collaborations in Pervasive Computing Environments. In I. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *The Semantic Web – ISWC 2006, 5th International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 473–486. Springer-Verlag, November 2006.

[22] A. Toninelli, R. Montanari, L. Kagal, and O. Lassila. Proteus: A Semantic Context-Aware Adaptive Policy Model. In *Eighth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07)*, pages 129–140. IEEE Computer Society, June 2007.

[23] M. van Kleek, M. Bernstein, D. R. Karger, and mc schraefel. Gui — phooey!: the case for text input. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2007. ACM.

[24] M. van Kleek and H. Shrobe. A Practical Activity Capture Framework for Personal, Lifetime User Modeling. In *User Modeling 2007*, volume 4511 of *Lecture Notes in Computer Science*. Springer Verlag, 2007.

[25] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3):94–104, Sept. 1991.

Acknowledements

The author is indebted to the following people for fruitful discussions and advice: Jamey Hicks, Deepali Khushraj, Mikko Perttunen, Alessandra Toninelli and Max "Electronic" van Kleek. The research described in this paper was supported in part by the Nokia Research Center.