DOWNSCALE 2013

Proceedings of the
Second International Workshop on
Downscaling the Semantic Web

Editors:
Victor de Boer, Martin Murillo and Christophe Guéret
v.de.boer@vu.nl, *murillo@ieee.org*, *christophe.gueret@dans.knaw.nl*

The Second international workshop on downscaling the Semantic Web was held on 19-9-2013 in Geneva, Switzerland and was co-located with the Open Knowledge Conference 2013. A call for papers was put out earlier that year. The workshop accepted three full papers after peer-review and featured five invited abstracts.

These proceedings introduce the issue of downscaling the Semantic Web, present the main outcomes of the workshop, and include the three full papers and five invited abstracts.

For more information visit http://worldwidesemanticweb.org/events/downscale2013/
Downscale2013 was sponsored by The Network Institute (http://www.thenetworkinstitute.org)

Table of Content

1. Introduction

Knowledge acquisition is a necessary and first condition for the empowerment of individuals. The need for appropriate and effective knowledge sharing is universal and global. Linked Data and Semantic technologies provide great potentials for carrying out those tasks. While mainstream Semantic Web research and development is moving vertiginously (focusing mainly on centralized and very powerful infrastructure and services in highly endowed application domains and regions where does not seem to be constraints), little work seems to be done on the applications of these and more appropriate technologies to less connected scenarios and challenged regions where new knowledge means day to day sustenance, survival, or the exercise of rights.

Indeed, 4 Billion people who don't have access to Internet or whose Internet connectivity is limited by bandwidth, quality of service, government or natural blockades, and modern device availability and affordability would welcome innovative solutions that are fit to their context and situation. The reality is that it will be tens of years until these subsets of the population enjoy the same level of Internet experience that most of the western population enjoys and takes it for granted.

Thus it is important to consider these stakeholders in the development of solutions that center around Linked Data. Additionally, in contrast to more conservative approaches that await various years to apply mainstream technology to the people that need the most, we believe that a parallel and cooperative approach with mainstream development can take place. For that purpose, we identify three major aspects that need to be addressed when bringing Linked Data to everyone: Infrastructure, interfaces, and content sharing.

Infrastructure: Current design of platforms and utilities that make use of Linked Data assume the availability of a Web infrastructure encompassing centralized data-centers, high speed reliable Internet connectivity, and powerful modern client devices. The implications can be serious: If any of these necessary conditions is missing, end users are unable to be served of the benefits that Linked Data provides. Naturally this is not only relevant in natural disaster scenarios but also in the reality of daily life of billions of people. Solutions that are less centralized and do not require constant connectivity are required, among others.

Interfaces: Literacy and language barriers currently prevent many people to reap the benefits of the World Wide Web, including knowledge acquisition, participation, and the exchange of ideas. Data-driven solutions such as Linked Data, being language-agnostic, provide huge potentials for the implementation of relevant interfaces for information sharing services, allowing more people to reap its benefits. Voice technologies, icon/symbol-based interfaces, touch interfaces, all provide unprecedented potentials, in the context of their power and lowering cost.

Sharing of appropriate content: Context and culture awareness are key for developing (Linked) Open Data applications. To ensure local uptake, it is paramount to identify relevant knowledge that is valuable to a community or a group, including local language to symbol appropriateness. While western-oriented approaches might seem globally applicable, the reality is otherwise. Linked data provides a huge potential in that context. This is not only applicable to the daily life of billions of people, but also in people's exercising their rights and taking part in democratic process with appropriate knowledge and greater potential for participation.

The workshop seeks to provide first steps in exploring appropriate requirements, technologies, processes and applications for the deployment of semantic Web technologies in constrained scenarios, taking into consideration local contexts. For instance, making Semantic Web platforms usable under limited computing power and limited access to Internet, with context-specific interfaces.

2. Workshop Outcome

The workshop was attended by nine people and one additional remote attendee. The workshop started with a keynote speech by Stephane Boyera[1]. Stephane highlighted the importance of knowledge sharing in remote rural areas in the developing world. He introduced a number of projects involving mobile and voice technology for knowledge sharing. The keynote speech was followed by a number of research talks and two roundtable discussions..

The roundtable discussions focused on the question what Downscaling actually is and identifying its different aspects/components. The consensus seemed to be that Downscaling involved the investigation and usage of Semantic Web technologies and Linked Data principles to allow for data, information and knowledge sharing in circumstances where 'mainstream' SW and LD is not feasible or simply does not work. These circumstances can be because of cultural, technical or physical limitations or because of natural or artificial limitations.

Figure 2 (below) illustrates a first attempt to come to a common architecture. It includes three aspects that need to be considered when thinking about data sharing in exceptional circumstances:

1. **Hardware/ Infrastructure**. This aspect includes issues with connectivity, low resource hardware, unavailability, etc.
2. **Interfaces**. This concerns the design and development of appropriate interfaces with respect to illiteracy of users or their specific usage. Building human-usable interfaces is a more general issue for Linked data.
3. **Pragmatic semantics.** Developing LD solutions that consider which information is relevant in which (cultural) circumstances is crucial to its success. This might include filtering of information etc.

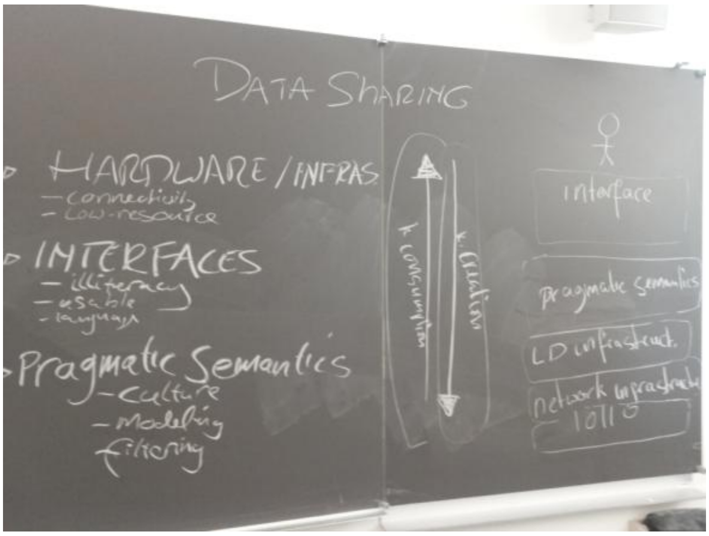
The right side of the picture illustrates the downscaling stack. On the top of this stack sits the human in the loop, the end use that wants to share data. On the bottom of the stack we can see the bits that make up the data. Downscaling is concerned with the levels of abstraction in between:

- The **Network infrastructure** deals with the need for solutions that deal with not-always-online networks, networks that might be very expensive to deploy and maintain, and networks whose bandwidth is not sufficient for mainstream data sharing solutions.
- The **Linked Data infrastructure** that is developed on top of this hardware also needs to take into account the limited connectivity of the layer beneath it. Furthermore we need to investigate how Linked Data principles and technologies such as URIs, RDF and SPARQL stay afloat in these situations.

[1] of SBC4D consultancy

- The pragmatic semantics concern the filtering of data that is to be distributed to the end-user (see above).
- The interface is crucial in allowing people in for example rural areas to share data (see above)

Another key dimension is shown in the middle part. Downscaling Linked Data should concern both the **creation** as well as the **consumption** of data.



Other important points to raise are:

- Downscaling focuses on the application of mainstream technologies to very constrained scenarios, the very same way that “appropriate technology” focused on different scenarios. However, given its parallel evolution, such technology can also provide valuable solutions when mainstream approaches can not work in some situations (natural disasters, failure of the grid, unstable nations, etc)

- A parallel approach is needed in order to make sure all stakeholders and all types of resources are considered; this goes anywhere from the conception of technical standards to the consideration of national and regional policies; i.e. some technologies, while being sound, can not be applied in some nations because of various issues

- Given the fact that ⅔ of world population do not enjoy the Internet and that there exists hundreds of ad-hoc independent approaches to putting solution to the consequences of this pressing problem, a unifying force is needed in order to provide focused, concerted and effective efforts that will gather the best out of all such ad-hoc approaches

The OLPC Project (One Laptop Per Child) : Recent Developments (2013)

Jean M. Thiéry

ModLibre.info
jean.thiery@olpc-france.info

Abstract

We present recent developments of the nonprofit OLPC project. The free-libre Sugar learning platform is now compatible with common GNU/Linux distributions such as Fedora, Mageia-Mandriva and Ubuntu. Recent XO laptops are more powerful but keep their original sturdy and low consumption design. OLPC tablets should be commercialized in 2013. Specific Sugar activities will complete already existing free (and sometimes libre) educational activities of Android.

Résumé

Nous présentons les développements récents du projet sans but lucratif OLPC. La plate-forme éducative libre Sugar est maintenant compatible avec les distributions GNU/Linux usuelles telles que Fedora, Mageia-Mandriva et Ubuntu. Les portables XO récents sont plus puissants mais gardent leur conception robuste et économe en énergie. Des tablettes OLPC devraient être commercialisées début 2013. Elles utiliseront le système Android en attendant l'adaptation de la plate-forme éducative libre Sugar. Des activités Sugar spécifiques compléteront les applications pédagogiques gratuites et parfois libres qui existent déjà sous Android.

Keywords

Education, Free-libre software, Laptop, Multilingual, Nonprofit, OLPC, Sugar, Tablet.

Mots Clés

Éducation, Logiciel libre, Multilingue, OLPC, Ordinateur portable, Sans but lucratif, Sugar, Tablette.

1 Introduction

The *OLPC* project (*One Laptop Per Child*) is based on more than 30 years of pedagogical researches. Since 2008, *Sugar Labs* (http://www.sugarlabs.org/) develop the specific Sugar environment and all its educational activities, while the *OLPC* foundation concentrates on the conception and promotion of dedicated XO laptops. Both Sugar Labs and the OLPC foundation are not-for-profit organizations. In this document, the expression *OLPC project* designates projects driven by both foundations or by local *OLPC* organizations (associations or public administrations).

An XO laptop may operate in conditions too difficult for usual netbooks : cold or heat, high humidity or dust, etc. All sockets are protected in closed position. Its keyboard is rubberized for protection against dust and liquids. Its screen operates in color mode or in monochrome mode (energy friendly and readable under bright light). The standard 12 V power supply is compatible with many electrical sources (dynamos, solar panels, ...) with storage on usual batteries.

With their Wi-Fi interface, XO's recognize each other and may be connected to neighbor computers (and available Wi-Fi hotspots) for collaborative activities of chidren with each other or with their teacher. Meshed networks may also be created. Local networks may include an XS school server optimized for XO's.

The Sugar learning platform respects the four liberties of free-libre software: execution, analysis, redistribution and improvement of programs. For an easier learning by young children, Sugar displays only one window at a time among four windows (neighborhood, friends, home and present activity). All activities are managed by a journal allowing to restart them in their last state.

The original OLPC-Sugar activities were the following :
- **Experiments**: Distance, Measure, Ruler, ...
- **Office and communication**: Browse, Calculate, Chat, Help, Paint, Record, Write, ...
- **Programing for all ages**: Etoys, Pippy (Python), Scratch, TamTam, TurtleArt, ...
- **System exploration**: Analyze, Log, Terminal, …
- **Teaching**: Moon, Physics, Speak, Wikipedia, ...

The OLPC project is multilingual. Activities can be easily translated into any language with a dedicated server (http://translate.sugarlabs.org/). This project collects only free cultural works (http://wiki.laptop.org/go/OLPC:License) and could play an important role in the transmission of all cultures (Thiéry 2009).

2 Evolution of the Sugar learning platform

The Sugar Labs foundation was created to make the Sugar learning platform compatible with different computers, in particular with second-hand computers usable for education.

Late 2009, the Sugar learning platform has been reorganized with a multilayer model for long term portability (http://wiki.sugarlabs.org/go/Taxonomy) :

- *Honey* : activities developed outside the *Sugar Labs,*
- *Fructose* : test activities optimized by the *Sugar Labs,*
- *Glucose* : the Sugar graphical environment,
- *Ribose* : the operating system and its interface with Glucose.

Sugar Labs have developed an autonomous version of Sugar called *Sugar on a Stick* (*SoaS* : http://wiki.sugarlabs.org/go/Sugar_on_a_Stick). *SoaS* can be tested on a live CD by downloading and burning an *iso* image (http://wiki.laptop.org/go/LiveCd). This image can be installed on a USB key (http://wiki.sugarlabs.org/go/Downloads). It may be executed in an emulator (QEMU …) or by virtualization (VirtualBox, Vmware, ...) on most systems, including Mac or Windows (http://wiki.sugarlabs.org/go/Supported_systems).

The Sugar learning platform is now available in the official repositories of most GNU/Linux distributions (http://wiki.sugarlabs.org/go/Downloads). Integration in Fedora is the best for historical and practical reasons : the first OLPC-Sugar versions were developed with this distribution which is still used for new developments (http://spins.fedoraproject.org/en/soas/).

Since 2011, Sugar Labs proposes the Sweets distribution which supports several GNU/Linux distributions without interfering with the rest of the system (e.g., it is possible to use Sugar from the Sweets Distribution and Sugar from official repositories at the same time) (http://wiki.sugarlabs.org/go/Sweets_Distribution). Sweets is the best approach to run Sugar activities with Ubuntu. It is also compatible with the single-board computer Raspberry pi (http://wiki.sugarlabs.org/go/Testing/Reports/Sweets_on_Raspberry_pi_armhf_raspbian).

3 XO laptop distribution

XO laptops are not sold directly to the general public. Reasons are complex and difficult to balance : in particular the multiplicity of keyboards fragments the potential *market*. Two *G1G1* initiatives (*Give 1 Get 1*) allowed to buy two XO's (one for an association and another one for the buyer). Distribution problems with these initiatives were not encouraging.

The normal XO distribution is done through public contracts between local authorities (town, region or country) and the manufacturer (e.g. the Taiwanese company Quanta Computer) with the possible competition of other solutions based on classical laptops.

Such a decision is expensive. Pilot experiments should be installed before

- to check the translation of all necessary activities (http://translate.sugarlabs.org),
- to prepare pedagogical documents compatible with national programs,
- to equip representative schools and
- to inform all potential decision makers.

The OLPC foundation may provide XO laptops for these pilot experiments but not for all cases. Requests for XO laptops should be well prepared and coordinated, if possible with the help of local OLPC associations.

Over 2.4 millions XO laptops have been distributed worldwide but mainly in Peru (870 000) and in Uruguay (380 000) (http://en.wikipedia.org/wiki/One_Laptop_per_Child). More detailed statistics may be found on the official site (http://one.laptop.org/map) or on the community map (http://www.olpcmap.net).

4 Future

The OLPC foundation prepares the specifications of future XO computers, following the classical approach of innovative projects proposing concepts, then prototypes, before mass production.

The XO-1.75 laptop was available at the end of 2011. It resembles the previous XO-1 and XO-1.5 but with more efficient and less expensive components (http://wiki.laptop.org/go/XO-1.75). It uses the ARM architecture (http://en.wikipedia.org/wiki/ARM_architecture) which predominates in embedded devices (mobiles, tablets, …) thanks to its lower consumption. Power consumption and purchase cost are both lower than for the XO-1.5. Changes in architecture has needed many changes in the deeper shells of Sugar (*Ribose* and *Glucose* : see above) but have less affected the upper shells (*Fructose* et *Honey*) mostly written in script languages (usually Python).

XO laptops are available with many types of keyboards for different languages. But these differences cannot be easily managed for a worldwide distribution. The OLPC foundation has conceived different projects with touch screens. The mechanical keyboard will be replaced by a virtual screen adaptable to all alphabets and national configurations.

The two-screen XO-2 project has been abandoned for cost reasons.

In 2011, the OLPC foundation and the Marwell company have studied a first tablet optimized for education, with a robust design and many connexions. The XO-3 tablet was announced in January 2012 but was not commercialized, probably due to the concurrence of many commercial tablets. For example, in France, some hypermarkets proposed up to 5 types of « educational » tablets at 99.90 € for the December 2012 sales ! 100 € seems to be a symbolic threshold for many child equipments (tablets, construction games, …).

A new OLPC tablet should be commercialized in the USA in the second quarter of 2013, probably with the Android system. The adaptation of the complete Sugar system to tablets cannot be done in few months.

Software is also an important issue for educational tablets. « While devices like eReaders and current tablets are terrific literary, media and entertainment platforms, they don't meet the needs of an educational model based on making things, versus just consuming them. Today's learning environments require robust platforms for computation, content creation and experimentation – and all that at a very low cost. » said Dr. Nicholas Negroponte, Founder and Chairman of One Laptop per Child (http://www.marvell.com/company/news/pressDetail.do?releaseID=1444).

The analysis of present tablets confirm this prediction. Upmarket tablets are practically closed : limited wired connections, unadaptable proprietary software and compulsory specific Web-stores. Most other tablets are based on the Android system optimized for mobile devices (tablets, smartphones, etc.). This system has been adopted by many manufacturers interested by its openness allowing all necessary adaptations. Android is partially free-libre since it is based on GNU/Linux and Java. Its openness in not guaranteed in the long term and each manufacturer tries to impose its Web-store.

In october 2012, more than 700,000 applications were available on Google Play, the Google general Web site (http://en.wikipedia.org/wiki/Google_Play). Many of them are free (no payment) ; some of them are free-libre and can be adapted to any use (e.g., translation into another language). The F-Droid Repository is a catalogue of easily-installable FOSS (Free and Open Source Software) applications for the Android platform (https://f-droid.org/). Most Android applications can be analyzed on a low-cost tablet or on a virtual machine like VirtualBox (http://www.virtualbox.org/).

Most Android applications are oriented toward culture, entertainment and specific services. Some Android Web-stores have no education category ! Available applications in education category are quite unequal : some applications are too simple, others are showrooms for commercial sites, while some of them have a real pedagogical interest. In March 2013, F-Droid presented around 600 FOSS applications, with 27 applications in the category « Science & Education » ! Other interesting educational applications could be found in other categories such as « Games » (68 games) and « Office » (126 general office applications).

The analysis of these applications is important since Android is already used by children, in particular on tablets or mobile phones. Many second-hand devices may be powerful enough !

Present tablets, with their best educational applications, prefigure the future OLPC tablet : the pedagogical interest of touchscreen, the easy localization of virtual keyboards, etc. However, a true learning platform cannot be obtained by juxtaposition of disparate applications. The Sugar platform with all its activities will transform the tablet use for education.

5 Conclusion

The new XO tablets are much awaited. Sugar activities for tablets can already be developed on personal computers and most of them may be tested on present tablets.

Acknowledgments

Thank you to all members of OLPC-France who have transmitted essential references and in particular to Bastien Guerry and Lionel Laské.

Reference

Thiéry J. M. (2009), Le projet OLPC (One Laptop Per Child) : un atout pour toutes les cultures, A*ctes de SITACAM'09*, 15-31.

Highly Available Entity Registry System for Poorly Connected Environments

Marat Charlaganov[1], Cristian Dinu[1], Teodor Macicas[2], Philippe Cudré-Mauroux[2], and Christophe Guéret[1]

[1] DANS, Royal Dutch Academy of Sciences, The Netherlands
{firstname.lastname}@dans.knaw.nl

[2] eXascale Infolab, University of Fribourg, Switzerland
{firstname.lastname}@unifr.ch

Abstract. Web of Objects and Linked Data applications often assume that connectivity to data repositories and entity resolution services are always available. This may not be a valid assumption in many cases. Indeed, there are about 4.5 billion people in the world who have no or limited Web access. Many data-driven applications may have a critical impact on the life of those people, but are inaccessible to those populations due to the architecture of today's data registries. In this paper, we propose a system that can be used as a general-purpose entity registry suitable for deployment in poorly connected or ad-hoc environments.

1 Introduction

Data registries are critical components of the Web architecture and are widely used in everyday Web activities [3]. For example, DNS registries contain registered Internet domain names, and allow Web users to use hostnames rather than IP addresses. Another example of registry is the Digital Object Architecture (DOA)[3]. It allows to assign unique identifiers to digital objects (e.g., scientific publications), which can then be used to access logical object properties even if their physical location may change.

In addition to those traditional registries offering hash-table like functionalities, online infrastructures and applications are increasingly turning to more flexible registries containing information about general objects or entities (i.e., object registries and entity registries) to power data-driven applications. Emerging examples of that trend are DBPedia[4], Freebase[5], and Wikidata[6] which, given an entity identifier, provide semi-structured metadata about the entity.

In situations where such registries are not continuously accessible, however, the user experience can be strongly limited. As a basic example, if the DNS server

[3] http://www.cnri.reston.va.us/doa.html
[4] http://dbpedia.org
[5] http://www.freebase.com
[6] http://www.wikidata.org

used by a client computer is not connected to the rest of the DNS hierarchy, then a very restricted set of Internet domains can be resolved to their IP addresses. In an object registry context, discontinued access to the registry typically results in the impossibility to publish data or issue object queries.

In this paper, we describe an Entity Registry System (ERS), which addresses the need to discover, connect, and exchange structured information in poorly-connected environments. In the absence of an Internet connection, ERS provides a way to discover and query other instances of ERS available in the local network, and transparently switches to querying the global server whenever the Internet connection becomes available again. Furthermore, ERS supports proactive replication of data between clients and the global server, local caching of data, global entity naming scheme, and provenance tracking, all of which simplify the creation of collaborative applications tolerant to connectivity failures.

The remainder of this paper is structured as follows. In section 2, we discuss a use case focusing of One Laptop Per Child "XO" laptops. Section **??** gives a high-level overview of our architecture. We compare our system to related projects in Section 4. Finally, we conclude and discuss future work in Section 5.

2 Use case: OLPC XO

The One Laptop Per Child (OLPC) initiative provides inexpensive rugged laptops (called "XO") to children in developing countries. XO laptops are used at school, but also at home where connection to the Internet typically can not be ensured. XO laptops automatically form mesh networks, such that data can still be shared between neighboring laptops in an ad-hoc fashion even if wide-area connectivity cannot be ensured.

For the sake of illustration, we consider an image-tagging game where children annotate pictures with keywords, and approve or reject tags suggested by classmates. A set of pictures is downloaded from the central server and distributed among children in the class. In the course of the game, tags and approvals are synchronized with the central server along with records concerning authorship. New contributions immediately become visible to the game participants. The same game continues after school sometimes without connection to the central server. Several children tag and review pictures in the same way as in class, but in this case they only see the data available on each other's laptops. On the following day when they return to school, the results of the offline game must be synchronized with the central server.

In this scenario, the ERS ensures storage for entity data, and provides an infrastructure for the discovery and the sharing of the data both in online and offline modes. Our choice of OLPC XO as the target platform for the initial implementation is motivated by the constraints imposed on the system design due to the limitations of XO's hardware, and due to the environment they operate in. However, the main components of our system are available on all major platforms since system as a whole is designed to be portable.

3 System Overview

The ERS API allows clients to store and retrieve RDF data at both entity and property level, e.g., to get all information about an entity, or to modify a property P for entity E respectively. Additionally, clients may search for entities using any combination of entity, property and value, i.e., to retrieve all data for an entity E having a given property-value combination $P - V$, etc.

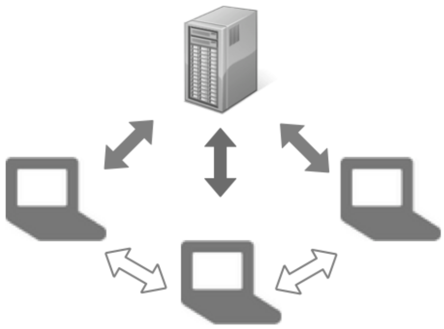


Fig. 1. Scheme of interactions in online (filled arrows) and offline (hollow arrows) modes.

Our proposed system architecture combines the client-server and peer-to-peer architectures by providing for an *online mode*, in which the machines can connect via the Internet to a cluster-hosted *global ERS component* and query the server, as well as an *offline mode*, for when machines have no internet access and have to use peer-to-peer techniques to retrieve data cached on the *local ERS components* across the mesh network. The ERS library switches between these modes automatically and seamlessly, so that the user uses the same API in both modes.

Provenance support is planned for the system, such that the source of every piece of information store in the system can be known and that the data can be filtered accordingly by clients to support collaborative edits, heterogeneous views, conflict-resolution, etc.

3.1 Global Component

The global repository provides a centralized space for storing all the global ERS data, and serves as a synchronization point for XO-laptops whenever their Internet connection is re-established. The context of this application requires the global repository to be a scalable, highly-available, low-latency store that provides AP or PA/EL consistency under the CAP [4] and PACELC [1] [2] models respectively. We have thus chosen Cassandra as a backend for our global component, owing to its maturity and proven track-record for performance.

CumulusRDF is an RDF store for cloud-based architectures [6]. It provides a REST-based API with CRUD (create, read, update, delete) operations to manage RDF data and uses Apache Cassandra as storage backend. This is the building block of our global repository storage engine. However, as we aim to bring lineage and to create graphs of entities, the API must be adapted accordingly.

In order to store triples on Cassandra and to be able to run different queries, cumulusRDF proposes two data models, one of which uses super columns. For each of them, explicit indices are also created to allow faster retrieval, thus allowing efficient queries by any (entity, property, value) combination. In order to support the concept of graphs, we have extended the cumulusRDF model such as to use separate Cassandra keyspaces (the equivalent of a 'database' in RDBMS terms) for storing data belonging to different graphs.

Performance is one of the main advantages of using Cassandra. One simple test of loading data using only one machine took around 2min for bulk-loading a graph with 5 million entities. We believe this is a reasonable result given our context. Moreover, query execution times vary only slightly with the size of data set, which indicates good scalability. However, a slightly longer response time is expected when using a cluster of instances, depending on the chosen replication factor and consistency level.

Replication and consistency are both tunable in Cassandra at the level of a keyspace or a graph. For a low-latency system, we may choose to sacrifice consistency and to either handle eventual inconsistency on the client, or to enforce read-my-writes for a better user experience. On the other hand, a k-replication varying from 3 to 5 is a safe compromise.

It would be interesting to provide entity-level consistency and replication. However, this feature is not available by default in Cassandra, thus further research and development must be undertaken. For the moment, we leave this as future work.

3.2 Local Component

Figure 2 shows the local components of the system and their interactions in local (i.e., "offline") mode. The following components are depicted on the picture:

- The ERS API, implemented as a Python package. The Sugar applications, which are the clients of the ERS system, interact with this library directly.
- The local entity data store, implemented using CouchDB, a document-oriented database.
- The Avahi daemon, used for peer discovery.

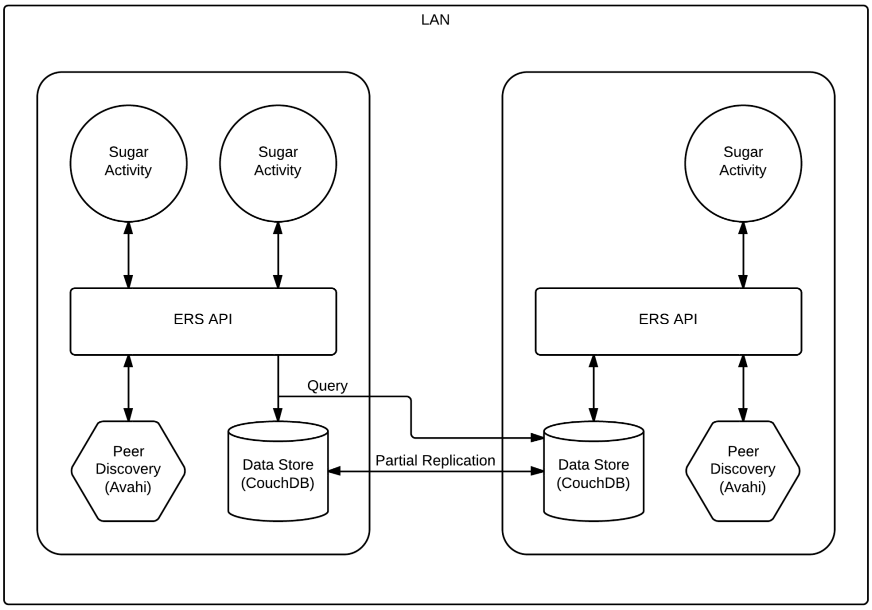


Fig. 2. The local system components

Local mode of operation Whereas in the normal 'online' mode all queries go to the global server, in local, 'offline' mode, the ERS library retrieves data stored locally on the client, as well as on its peers, on a best-effort basis (see figure 1). The library achieves this by sending queries to both the local CouchDB instance, and to the CouchDB instances on any known peers (which are also accessible remotely via HTTP), and then merging the results.

The CouchDB data store The local CouchDB instance on every client is responsible for caching entity data, as well as storing any other stateful information for ERS operation. CouchDB is a key-value store that supports arbitrary JSON data as values (called 'documents'), map-reduce operations, secondary indexes, and replication. We chose it for its flexible data model, advanced facilities and excellent support for concurrency.

A key issue in our design was the selection of a mapping and indexing model for storing entity data in CouchDB in order to enable efficient queries and replication. We are currently experimenting with several models and have designed the library such that we can switch seamlessly to any specific model.

CouchDB supports automatic replication, which is used to propagate data through the network as well as any other piece of information. Although replication relationships have to be explicitly established, the library handles this by using data from Avahi.

Peer discovery The ERS API interacts with the standard Avahi daemon in order to keep track of the peers nearby. Although Avahi cannot detect peers across multiple hops, this is sufficient for the typical usage scenario (kids in a classroom), and data from remote peers eventually propagate via replication.

4 Related Work

The Sugar Network project[7] addresses the challenge of data sharing and collaboration in environments with intermittent Internet access by introducing node servers to the local area network. Node servers synchronize with each other and with the master server via sneakernet protocols or via network when the connection is available. A notable difference with our approach is a pre-defined schema which limits the types of entities that can be shared.

Our approach follows several ideas introduced in the SemanticXO [5] project. SemanticXO adopts a flexible RDF data model to enable data sharing in a local area network. The software exposes entity information via a SPARQL endpoint and provides resilient means for querying data available in the network.

5 Conclusions

We have discussed how entity registries lie at the core of many interesting Linked Data applications that can help the poorly-connected parts of our world. As traditional solutions are unusable in such a context due to their fundamental design assumptions (such as powerful hardware or a permanent connection to the Internet), there is a need for a downscaled entity registry solution that can work in an exceptionally adverse hardware and connectivity context.

Our solution, dubbed ERS, is designed from the start with this particular context in mind, and as such is highly resilient in the face of the issues associated with a low connectivity environment. At the same time, our solution is generic enough that it can be used as a fundamental component for a wide variety of applications, from the educational to the social and beyond. We thus have the expectation that Sugar application developers will benefit from our system and be helped in the development of solutions that can improve the life of otherwise isolated communities in poorly developed parts of the world.

Planned future work includes provenance support, benchmarking and selecting CouchDB/Cassandra data models, as well as research towards intelligent replication and local-global synchronization.

6 Acknowledgment

This work was supported by the Verisign Internet Infrastructure Grant program 2012.

[7] `http://wiki.sugarlabs.org/go/Sugar_Network`

References

1. Daniel Abadi. Consistency tradeoffs in modern distributed database system design: Cap is only part of the story. *Computer*, 45(2):37–42, 2012.
2. Eric Brewer. Cap twelve years later: How the rules have changes. *Computer*, 45(2):23–29, 2012.
3. Philippe Cudré-Mauroux, Gianluca Demartini, Christophe Guéret, and Benoit Perroud. Downscaling Entity Registries for Ad-Hoc Environments. In *Proceedings of the 1st International Workshop on Downscaling the Semantic Web*, 2012.
4. Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent available partition-tolerant web services. ACM SIGACT News, Volume 33 Issue 2 (2002), pg. 51-59.
5. Christophe Guret and Stefan Schlobach. Semanticxo: Connecting the xo with the worlds largest information network. In JimJames Yonazi, Eliamani Sedoyeka, Ezendu Ariwa, and Eyas El-Qawasmeh, editors, *e-Technologies and Networks for Development*, volume 171 of *Communications in Computer and Information Science*, pages 261–275. Springer Berlin Heidelberg, 2011.
6. Günter Ladwig and Andreas Harth. Cumulusrdf: Linked data management on nested key-value stores. In *The 7th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2011)*, page 30, 2011.

„How It Grows?“

Linked Open Data as a management reference when monitoring production processes in rural communities or small industries using a Wireless Mesh Network controlled by a Raspberry Pi

Christophe Dupriez

DESTIN Informatique, Felenne, Belgium
dupriez@destin.be http://www.destin-informatique.com

Abstract. This project addresses the needs of small industries or farmers and forest managers in rural communities. It may interest also other types of operators who need to control multiple remote processes. Resilient Mesh Networks major advantages are small size, small cost, small energy, resilience and expandability (but small speed: no video or sound). We experimented Telegesis implementation of Ember System On a Chip (which combines an ARM CPU with Zigbee hardware+firmware and Analog/Digital converters + I/O ports) together with analog sensors and relays for very polyvalent and low cost field modules. A very low cost Raspberry Pi is used as a network controller and for the management application. Data will be stored "on the fly" as time stamped RDF triples in a database. Different options to collect reference data from Linked Open Data (LOD), to build the management application and to publish real time statistics on the LOD are currently in development.

Keywords. Raspberry PI, Telegesis, Analog/Digital converters, Monitoring, Farms, Forest, Linked Open Data, Radio-transmission, Wireless Mesh networks, Rural communities, Small industries, Remote sensing

1 Introduction

This project addresses the needs of small industries or farmers and forest managers in rural communities. It may interest also other types of operators who need to control multiple remote processes. In this paper, "growers" is used to designate all of them.

The one who has the leisure to grow a garden, is looking at it every day: (s)he knows what is happening.

But the one trying to earn a living must monitor many things at a much greater scale: rational use of inputs, production quality, conservation, etc. despite numerous imponderables.

The grower may like to react to every event as they come or prefer patient observation before investing. (S)he may also prefer a middle term: to learn in the context of action, ensuring that first investments are sufficiently modular to be reused thereafter.

Unfortunately, commercial solutions are often focused on a very specific application where the cost / benefit ratio may be praised by the reseller. They are rarely modular, open-source and interoperable: they aim to force the grower to continue investing either in overvalued add-ons or in completely separate solutions.

Users are left with a plethora of incompatible applications (*Numérique et agriculture : quelle réalité de terrain?* Christopher Castro http://www.inriality.fr/agriculture/usages/aquitaine/numerique-et-agriculture/ : Interview of Karine Breton, Observatory for Agriculture and ICT (Aquitaine, France): "*People are equipped with many different software. A winemaker, for example, may have software for each of his/her activities (cellar management, winemaking, winery management, Internet online sales, CAP statements,* *etc. Same situation for a breeder (declaration of birth, death, animal origin certification ...). Everyday, too much of the same data must be entered in these different software. Furthermore, there is an interoperability problem between them.*")

The flexibility of modularity has a cost (for a similar budget, so many differences between a Spartan & modular Citroën 2CV compared to a C1 comfortable but fragile!). The modularity also implies that the grower understands each of the modules and how they can recombine (or not) differently: these skills must be acquired and a large community of users (nearby or accessible through the Internet) is required to exchange advice and experiences.

A large community of users is also important for the durability of the solution: nothing worse than an orphan solution whose spare parts are becoming scarce and expensive.

These observations lead us to develop *systems* (sets of *reusable* components assembled with a specific objective) modular, open-source and interoperable, based on standards well recognized and supported by large communities.

2 Project Architecture

The "How It Grows?" (HIG) project aims to bring fields' data to the living room (or to the smartphone) so the grower does not listen only "His Master's Voice" but also the voice of his/her various plots and buildings: is it too hot? too thirsty? Does it need watering? Is it too dark? Light up one of the spots? Is the room too poorly ventilated? invaded by insects? Do Animals eat enough? Does the tank have enough water? too much? etc.

The information has no more value than the savings from the actions it triggers. The transmitted information can trigger human action, it can also trigger controllers:

- To start / stop watering with a sprinkler
- To start / stop heating
- To turn on / off lights
- To open / close a fan, a ventilation
- To send food to plants or animals
- To open / close valves
- etc.

To measure data from different locations, it is necessary to collect it (sensors) and to transmit it. Simple modules (minimal settings) must be installed everywhere needed (and thus be very cheap) and transfer the information to a central control station which must perform storage, analysis and display to users.

Analog sensors, connected to the small modules, are mostly inexpensive and measure:

- temperature (air, water, body)
- brightness (external, internal)
- humidity (outdoor, indoor)
- atmospheric pressure
- pH (soil, hydroponic plant food, etc..)
- CO_2 concentration (level of animal activity, quality of ventilation)
- presence (animal intruders)
- wind (direction, force)
- vibration (insects into a trap)
- noise (activity)
- distance or level (ultrasonic sensor)
- acceleration
- pressure point on a surface (position)
- stretching (weight measurement)
- etc. (many proposals from GROVE system: http://www.seeedstudio.com/wiki/GROVE_System)

To those simple sensors, it must be possible to add more sophisticated modules that work locally in a short feedback loop. For example, to measure the weight of a bird and the amount of food it eats:

1. weight the manger,
2. detect the arrival of the bird,

3. weight the bird on the manger,
4. detect the departure of the bird,
5. immediately reweight the manger.

Transmissions from modules to the central controller (and vice versa) must be reliable and able to reach all locations of the farm (enterprise).

3 Hardware Experiments

The wireless network topology we choose is the Mesh where nodes retransmit messages successively up to the central station (the “controller”). Transmission standards like "ZigBee" (http://www.zigbee.org/) seem very attractive on paper for an array of reasons:

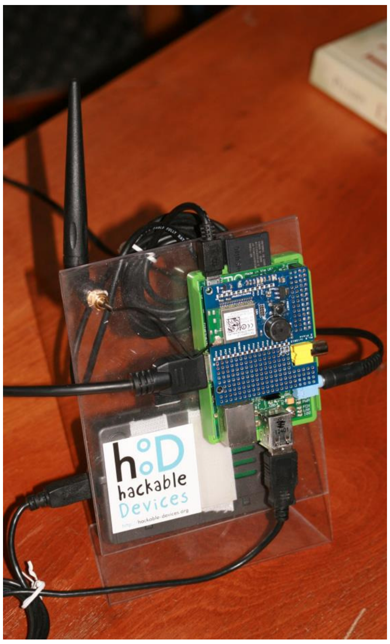
- Standard followed by many manufacturers and ensuring that the components are interchangeable (within the limits of the functions covered by the standard);
- Very low cost: 20€ for a 5cm2 module combining analog measurements, transmission according to the ZigBee standard (2.4GHz), able to control relays, LEDs and buzzer;
- 80 meters effective distance for communication between modules (200 meters without obstacles);
- Mesh network with up to 30 relays ("hops") possible. Messages can therefore travel a total distance of more than two kilometers ;
- Operates on simple AA batteries for more than one year;
- Reliable: ZigBee is also used for patient monitoring: http://docs.zigbee.org/zigbee-docs/dcn/09-4962.pdf

For the field module, we began our developments with one produced by the British firm Telegesis which is based on the ZigBee System on a Chip made by Ember (http://www.telegesis.com/downloads/general/Telegesis%20Product%20Range%20brochure.pdf).

This module is controlled by directives similar to Hayes modems ("AT" commands).

The modules must be powered and connected to sensors and relays: for this, we use the Z-Shield (http://www.bitbang.co.uk/zshield.html), a "Shield" (stackable circuit board) for the "open hardware" microcontroller Arduino (but we do not use an Arduino board below the Z-Shield). The Z-Shield has sensors for brightness and temperature, a "buzzer" and two LEDs. It has plenty of room to install a connector suitable for analog sensors, relays, etc.

"Arduino" open and free (like in “freedom”) hardware+software is supported by a huge community (hundred thousands enthusiasts): if an application requests a tight local control loop or a digital sensor (not just an analog one), we can just put an Arduino compatible board under the Z-Shield.



BitBang.co.uk also produces the Z-plate (http://www.bitbang.co.uk/zplate.html), a board stacking over a Raspberry Pi (http://www.raspberrypi.org) with characteristics similar to the Z-Shield. We therefore use a Raspberry Pi as a central network controller: it costs very little (35 €), consumes little (less than 10 watts) and can stay on all the time. Raspberry Pi is supported by a huge community (more than a million were already sold end of 2012). RaspberryPi is the Citroën 2CV of computing: it is not speeding but copes with most chores. One does not hesitate to replace it if some failed test blew it! Furthermore Oracle uses Raspberry Pi for its Java implementation tests on ARM CPU: Java 8 works very well on this platform.

Field experiments showed:

- Effective distance is nearer from 40 meters than 80;
- Low power operation ask for end devices sleeping for at least 10 minutes between each measurement: even then, AA batteries are exhausted after a few months;
- End devices cannot sleep for more than 2 minutes without being removed from the network;

- "AT commands" are not efficient for a network where the nodes sleep and get awaken periodically: the sequence of operations to do and to report cannot be piloted remotely, it must be programmed locally in each node;
- Radio modules are closed hardware and software (they include a user license for the Wireless Network software: ZigBee trademark is subject to a "membership fee" that manufacturers need to recover): it is not possible to add local applications within them except by adding a second microcontroller;
- 2.4Ghz may not be the best radio bandwidth for applications needing only a low data bandwidth (conflict with Wifi, bad traversal of walls).

We tested many other deceptive possibilities from April 2013 up to now (September 2013): we will report as soon as experimentation confirms something suitable. Our current hopes are in the DigiMesh 868MHz modules for which we had to produce a Surface Mount to Through Holes adapter even before testing them.

4 Packaging and Connectors

This application works outdoor: packaging of modules and their connections are an important issue.

Some modules can be in rainwater tanks: it is obvious that the packaging will be adapted (starting from an outdoor electrical box) with simple fixed connections.

But most of the modules will be more accessible and must be easily adaptable. The packaging and the connectors can quickly cost more than electronics: we propose to use a very simple package ("roof" made of a metal sheet, frame made from varnished wood, Velcro for the different fixations) and flexible connectivity based on the most popular standards (mini-stereo audio for analog sensors = Ground + 3.3volts + signal voltage (between 0 and 3.3volts) to be measured, RCA jacks for relays = Ground + 3.3 volts / 4 or 8ma output signal). It remains to determine the best combination of connections within a width of 10 cm (for instance, 2 analog inputs, 3 on/off input signals, 3 outputs) knowing that temperature and lighting are already measured onboard and that we can make additions for specific needs.

5 Technical Choices (Software)

We did not think necessary to invest first in the development of software for the fields' modules: we hoped to confine ourselves to what "AT" commands would allow.

But this is not sufficient in different cases:

1. Digital sensors (for instance sonar, high precision GPS) must be read by a serial port (SPI, I2C, TTL, USB, etc..) and require intelligence to be interfaced: an additional microcontroller (and more battery power) is then needed with some control software.
2. Control applications may require following a rapid succession of events: they need local monitoring and therefore a local microcontroller too.

We first focus our development efforts in the Java 8 application programming on the RaspberryPi: this first application only receives and archives network data without processing. When sensors calibration is improved (or any other parameter), this archiving allows to reinterpret with the new calibration the raw data received earlier.

This first level is a simple Java application that listens to the network and archive events received following a very simple conceptual data model (CDM):

1. **about**: sequential numerical identifier for the measure
2. **timestamp**: date / time / seconds / milliseconds of the measure
3. **network**: identifier of the telecommunications network
4. **module**: identifier of the module that made the measurement
5. **register**: identifier of the register within this module
6. **value**: measured value

The setting of this application will be based on pattern matching: regular expressions (Java RegEx) will be applied on the message lines received from the network.

We were thinking to use a standard database for on the fly logging of the data. Looking at the load it would impose to the Raspberry PI, we decided to investigate the RRD (http://oss.oetiker.ch/rrdtool/) and more precisely to its Java incarnation, RRD4J (http://code.google.com/p/rrd4j/). JRDS, a Java viewer of the chronological data series also exists (JRDS.org): it has been recently adapted to our needs (data sent by the network and not requested by the central monitoring application).

Access in RDF format for the collected information still has to be developed: the URL of each measure will be created using the sequential numeric identifier "**about**" and with the different properties mentioned in bold above.

The application should monitor failing modules (modules that do not send their information in due time) and trigger re-init sequences from time to time (failures will be logged and signaled).

At a second level, a set of categorization rules will take raw data and turn it into usable data for analysis and presentation. The rules have validity intervals over time to take into account changes to connected sensors (a module register could be linked to a temperature sensor up to a certain point and after be linked to a CO2 sensor). SPARQL is a potential solution to parameterize this level but we will start with SQL.

All the parameters will be stored in a catalog managed using open access repository software DSpace.org together with ASKOSI.org (storage / publishing of information according to the standard SKOS / RDF). This catalog will contain the definitions of:

- networks (with their types)
- places (sites, buildings, ...)
- modules (with their types)
- types of sensors (with their calibration method)
- measures (register in a module in a network with its calibration rule for a period of time)
- data interpretation rules to create events (for instance "alert if it is too hot in the barn")
- external reference data (AGROVOC in the languages of grower, DBpedia information related to the activities of the grower, etc..) will be used to contextualize the settings (for instance, optimal temperatures for tomatoes depending on their maturity).

The events created by the rules of interpretation will also be added and indexed (keywords to describe the events) in DSpace: retrospective searches are therefore possible.

For each object type in DSpace (network, location, module, position, event), a timeline with associated data will be automatically displayed (existing JavaScript library). This timeline will show the detected events, the minima, the maxima and the accruals (where relevant). Each point in the timeline allows access to the underlying information.

The publication of this information in RDF will contain a property "source": this will link to measures used to calculate each information published.

6 Reference data

The "How it grows?" (HIG) project shall identify reference information about the sensors, calibration, plants, animals (their environmental constraints), etc. It will be found in the "Linked Open Data Web" or otherwise the project will have to produce and publish them. Dynamic weather data will be certainly very useful too.

The community should be structured so that this data collection / creation process is managed properly.

All information will be accessible through a SPARQL server and the data will be stored locally (periodic refresh) to allow proper functioning even during Internet access failures.

7 Published Data

This project is not a "Big Data" project: we do not try to know everything about growers except for what they do want to share. Sites like http://Cosm.com and most home automation applications store data for free on their servers and make them available on the Internet. But they can freely reuse the collected data for their own aims and they often need a way to “monetize”.

We therefore propose that the application is made available autonomously on the Internet (even when dynamic IP) in three possible ways:

1. by an intermediate server providing a dashboard limited to important alarms and synthetic data but without storage;
2. by publishing selected Linked Open Data in a SPARQL / RDF server managed by the HIG project or another;
3. by using a dynamic DNS and a local router configured with a DMZ (and/or VPN) for external access by the grower to the entire application.

8 Conclusion

The "How It Grows?" (HIG) project has stills many proofs to make. We hope to develop it in the spirit of rural communities (resources invested sparingly, autonomy of every actor, sense of the "chore" to gather the community and do what is necessary: http://www.shareable.net/blog/farmhack-collaboratively-retooling-agriculture) and to be in close relation with various rural or small industry stakeholders (farmers, agronomists, planners, local authorities, regional agencies, quality auditors, computer developers, manufacturers ...). Comments and suggestions from the community are very welcome!

Semantic Stream Processing for Resource Limited Devices

Colin Hardy, Wei Tai, Declan O'Sullivan

Extended Abstract

To increase the interoperability and accessibility of data in sensor-rich systems, there is a proliferation recently of using Semantic Web technologies within sensor-rich systems such as hazard monitoring and rescue, context-aware computing, environmental monitoring, field studies, internet of things, and so on. These applications often need to process data streams received from sensors. A common paradigm is that sensors transmit all raw data streams to a central node in which all processing such as semantic uplifting, data fusion resides. This pure centralized paradigm can leverage the powerful processing ability of a central node. However, it has several drawbacks. The increase in both the number and types of sensors can enormously increase the amount and complexity of data, therefore requiring often upgrading the existing infrastructure (network and central server). More importantly, deploying all processing capabilities on a powerful central node relies on reliable and fast network connections, which is however can be difficult to achieve in a hostile environment. This is in particular the case for subsea sensors (e.g. seismometers, acoustic surveillance sensors), where existing data collection mainly relies on in situ data collection (requiring travel into the field by scientists), undersea optical cables or limited (10kbps at a range of 1km to 10 km) underwater wireless communications. Shifting some data processing on to the sensors themselves is increasingly becoming of interest.

In our research, we have built a prototypical semantic stream processor, called "stream Coror" (see Figure 1).

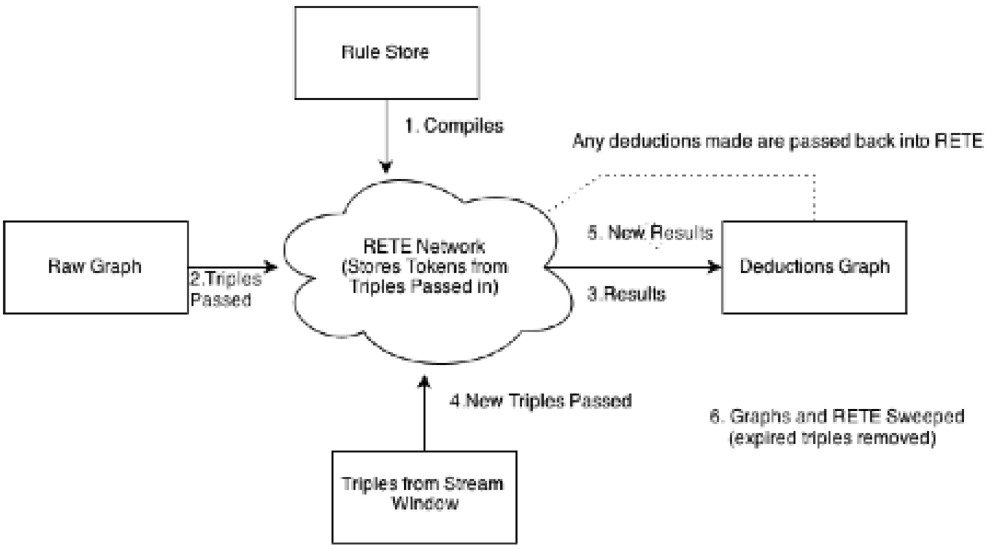


Figure 1: Stream Coror

This stream processor receives sensor measurements as a RDF stream and performs realtime reasoning over the data for resource limited environments. Ontology inference rules are used to reason the data streams against ontologies specifying static information such as data models or topology to infer implicit knowledge and to detect semantic inconsistencies in the data. Domain-

/application- specific rules can also be added to perform (simple) data fusion in realtime so that only meaningful results and results that are of interest are produced. In this way some data processing burden can be shifted to sensor nodes, decreasing a system's reliance on fast network connection and a powerful central processing node. The stream processor uses a rule-based approach for ontology reasoning and a sliding window method for handling streams.

Various variables were altered in our experiments in order to see how the reasoner would react. Trends were seen for triples added to the ontology from the stream window, window size and also ontology size. Well defined linear trends were drawn from each of these variables. These trends can help to predict the reasoners capabilities, and to help others in order to compare their implementations to this reasoner. Current work focuses on improving performance by reducing searching for triples to be removed from window processing through the use of indexing techniques.

Sheabutter and Knowledge Sharing in rural Africa

Anna Bon
a.bon@vu.nl

Women farmers in Tominian , Mali produce pure, high quality sheabutter. They offer it at 1000 fCFA per liter (1,50 EUR). Unfortunately, nobody buys it. Low quality, smelly sheabutter is sold at the local markets for 500 cFA.

The women want to increase sales. But how to pack sheabutter in small quantities? How to keep sheabutter fresh in the heat? A laboratory studied the composition of their sheabutter, but there are no standardized quality measures for sheabutter. Who are other producing regions of sheabutter? Which process do they used? How do customers use sheabutter? Which recipes do customers use when baking with sheabutter?

Some answers could be found on the web e.g. Wikipedia, or on best practices websites from NGOs. These women do not read. Besides, they cannot share their best own practices, because they cannot write.

There is much knowledge in rural Africa. The web still does not provide modalities to enter voice data, nor access data using voice/mobile phone. There is no structure to store or search best-practices in voice or e.g. video.

The sheabutter is one case, but illustrates the general problem in rural Africa. A solution is needed!

A Dialogue with Linked Data: Voice-based Access to Market Data in the Sahel

Victor de Boer
v.de.boer@vu.nl

The Linked Data movement has facilitated efficient data sharing in many domains. However, people in rural developing areas are mostly left out. Lack of relevant content and suitable interfaces prohibit potential users in rural communities to produce and consume Linked Data. In this paper, we present a case study exposing locally produced market data as Linked Data, which shows that Linked Data can be meaningful in a rural, development context. We present a way of enriching the market data with voice labels, allowing for the development of applications that (re-)use the data in voice-based applications. Finally, we present a prototype demonstrator that provides access to this linked market data through a voice interface, accessible to first generation mobile phones.

Challenges and opportunities in downscaling services in extreme remote villages in the Amazon

Martin Murillo
IEEE - Humanitarian Technology Challenge

In the absence of fiber optic or other cable-based connectivity, some remote areas of the world enjoy Internet connectivity through satellite or long-distance point-to-point systems. However, because of weather, technical issues, lack of payment, etc Internet gets cut for different periods that might be in the minute, hour, day, or month range. In this situation, high-speed connectivity between these remote areas are generally available, although with no Internet. Additionally, these areas do not enjoy grid electricity and they are limited to only hours of electricity every night. In this context, villagers need to get the most out of their connectivity through:

1) Communicating between rural health offices at anytime of the day (note that Internet is not available but links among various points are available)
2) Be able to access locally or remotely stored information anytime of the day
3) Make such content useful in local context: i.e. statistics on patient diseases/issues, etc
4) Be able to troubleshoot problems and to report to central point possible issues

In this presentation, we provide a case study of this very situation and explore on how downscaling services can contribute in solving the most pressing issues.

Towards open and semantic representation of a monitoring platform in the Peruvian Amazon

River Quispe Tacas, Cesar Córdoba
Rural Telecommunications Group
Pontifical Catholic University of Peru

The towns, villages and cities located in the Peruvian Amazon basin, as a result of the almost total lack of roads, depend mostly on the navigability of the rivers for the daily survival of the inhabitants and for the development of commercial activities (transport of goods and people). The presence or absence of rain dramatically affects the transport of goods of poor farmers and the decision on whether sick people should be evacuated in cases of emergency. The river can be non-navigable (the river is below the minimum level of navigation) or can provide optimum mobility, depending on the level of water.

In order to improve remote villager's decisions on whether to travel or not (i.e. evacuate or not), the rate of change of river level can be determined along several key points of the river. This would allow to lower uncertainty on the decisions on whether to navigate and avoid being stranded; this can at times be deadly or introduce high costs to the poor citizens.

The current project consists of a remote monitoring station that measures water quality and water level in the Napo River. The river is located in the town of Santa Clotilde, 212 km from the city of Iquitos, a main hub for the Peruvian and Brazilian Amazon. The monitoring station is composed of sensor nodes located in Santa Clotilde and an information system located in the Regional Hospital of Iquitos. We utilize point-to-point communication to relay the information from the sensors to the central station located in the hospital.

More specifically, the parameters monitored are temperature, dissolved oxygen, turbidity, acidity, and the level of the river. The data is sampled each hour and sent to the central system where is stored in a relational database and displayed graphically on the web. Given the potential of data of being combined with other datasets to provide further meaningful and valuable information, we are in the plans to publish such real-time measured information in open data formats such as RDF (i.e. through a "RDF Water Quality"). We also plan to introduce semantic web to the datasets in order for the data to be interpreted by the various villagers that live in different villages.