SAEON Knowledge Repository

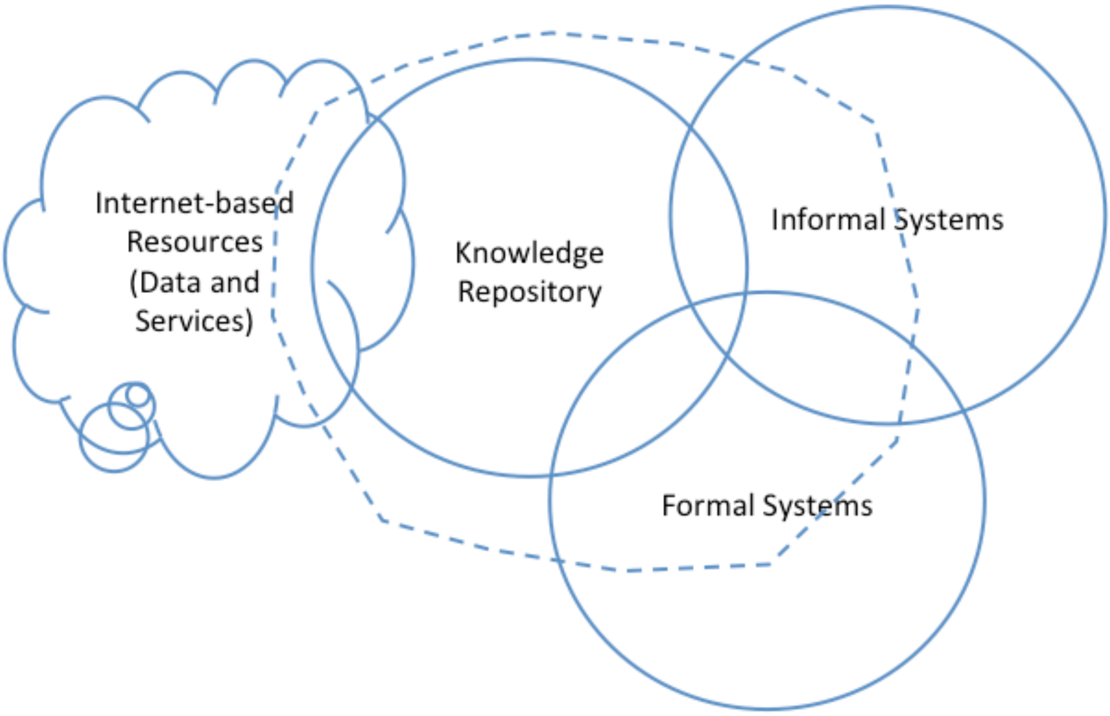
Concept
W Hugo
Version 1 – February 2014
Draft

Why do we need a Knowledge Repository?

There are many reasons, but two compelling ones:

1. A Business Case:
   a. Due diligence in managing and preserving evidence of performance specifically, and in general tracking our collective output.
   b. Improvement in efficiency and saving time.
2. An Information Management Case: Operating on the 'Single Source of Truth' principle – by limiting the number of times that information is stored and duplicated.

Success is measured in reduction of the time and effort needed to find the right outputs and to report on them.



The scope of the knowledge repository includes formal systems, less formal systems, and digital object repositories, and the edge of the repository is fuzzy: not all its resources are stored within SAEON's own systems.

A Variety of Information Objects

Any organization today generates information across a wide variety of platforms and in a wide variety of formats. In SAEON's case specifically, these range from formal, structured systems to very informal, poorly structured pieces of information. **Annexure A** provides a detailed list of the typical 'families' of information object that we need to deal with.

Not only are the objects distributed across systems and formats, but they are also subject to very different modes of control: some of the information is directly under the control of SAEON, some of it is controlled by our host organisations (NRF, SANPARKS, etc.), and some are controlled by entities completely unattached to us (for example, the publications in a peer reviewed journal). All of this variety needs to be accommodated in a consistent way.

The challenge is, then, to extract the maximum amount of information and knowledge from this disparate range of objects, and to do so in a way that is not complicated or difficult to maintain.

Flexible, Open Design

To achieve this, we require a flexible, open design. It should be flexible, in the sense that we can easily accommodate new types of objects in our system without a large amount of rework. It should be open, in the sense that it is possible to apply many tools and applications directly on our knowledge base without too much work.

The first objective, ***flexibility***, is largely achievable by allowing non-formal objects (i.e. information objects that are not stored in a formal, structured database) to be ***tagged*** more or less as we wish. It is important to understand that this flexible tagging achieves two things:

1. It allows multiple filing structures to be imposed on an information object. Many electronic filing systems are one-dimensional and rigid, in the sense that there is only one 'filing structure', and this is often not easy to extend. This has major failings, not least because most information objects need to be filed in more than one context (See **Annexure B** for a discussion on the ***dimensions*** of our corporate environment).
2. Secondly, associating tags with information objects is the first step towards constructing a ***knowledge network***, in which information objects are related to one another by way of one or more formal relationships. See **Annexure D** for more.

We should insist on a minimum number of critical tags, but ***we should not limit users*** to these tags. This means that any user can apply additional tags, as needed, to the information objects.

The objects that reside in structured databases and systems are already tagged: we must just make sure that their tags are (largely) the same as those used to tag informal objects.

Secondly, in terms of 'openness', we need to allow for the following:

1. Our knowledge repository will consist of a large number of formal and informal objects, linked together in multiple relationships based on several mandatory and optional 'filing systems' (collections of tags). This repository should allow query and interrogation in standard ways, or provide tools that allow the outputs to be available in a number of standard formats. **Annexure C** contains some details on our thinking in this regard.
2. Although the system should be open, and allow maximum exposure of items to users, it must also honour restrictions on access and modifiability.

Re-Using What We Have

SAEON has two important platforms that we intend to re-use as the major components of our Knowledge Repository:

1. The Shared Platform, that houses the SAEON Data Portal amongst others, can easily be extended to serve as the primary portal for our Knowledge Repository. This has already been done, and the beta version can be found at http://skp.dirisa.org . This platform has many advantages: It already takes care of publication life cycles, searches and discovery, visibility, ownership and sharing, services for reporting, and accommodates a wide variety of standard digital objects.
2. Development on a 'BigTable' open relational database system – capable of accepting user-defined, forms-based small systems with indeterminate structure. This facility has many generic tools – including built-in spatial awareness, query and reporting, filtering, and so on. This tool is eminently suited for small management systems, such as asset management, program management, and maintenance of master lists of data. An example implementation can be found here.[1]

The Central Concept: Stubs

It is clear from the earlier discussion that the central problem revolves around building linkages between information objects that can be anywhere in the web, including those in our portal and content management environments, and the formal systems and master lists that define our critical corporate dimensions and 'filing systems'.

We need to accommodate a number of typical ***use cases***:

[1] http://app01.saeon.ac.za/PLATFORM_3/MAP/capture.asp?CAMPID=83

1. An information object is stored externally to the organization (the most common example being a peer-reviewed article published on a journal website and stored in their repository). We need to tag this object in respect of persons that authored it, possibly the node that should be credited with it, the KPI it applies to, the projects, grants, and collaborators that were involved.
2. An information object is stored within one of our internal content management portals (such as the corporate site, the Knowledge Repository, the SAEON Data Portal). An example could be a meta-data record, a biographical sketch of a researcher, a news item, or information on a project. These objects need to be tagged in much the same way as above.
3. An information object that resides in a formal or semi-formal system. These are less simple to reference to a set of tags, since the system must typically be amended to allow for such tagging. An example may be the properties and attributes of a formally documented project, information about assets, or the details of a person registered in the HR system.
4. An information system that we use – but is outside of our control. Examples include the management of grants in RIMS, and the soon-to-be-implemented ERDMS. These are very difficult to tag: it requires an outside organization to make adjustments to their system.

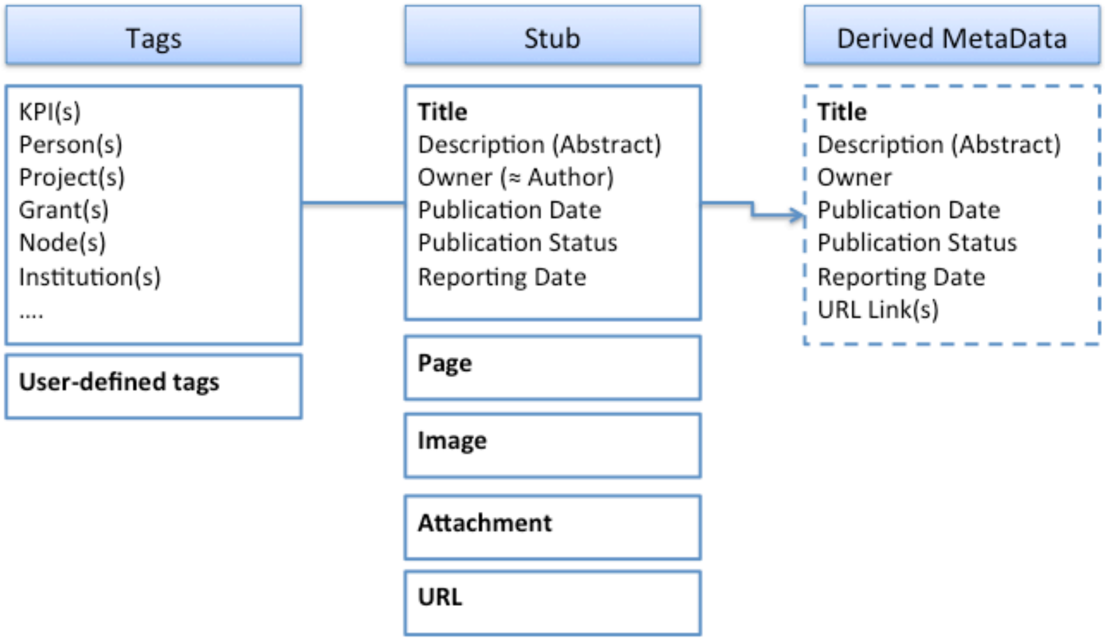
Stubs, together with our BigTable system capabilities, allow us to bridge the gaps – especially in respect of cases 3 and 4.

Hence our strategy is:

1. Supplement formal systems (such as HR, Finance, and RIMS) with informal BigTable systems that share a common key. This provides both information objects and some of the dimensions that we need for tagging.
2. Complete the set of tags and dimensions that we require by creating BigTable systems for them.
3. Compile services for tagging that draw on both for the purpose of classifying information objects.
4. Allow users to create objects as and when they need it as stubs, or any other object allowed by our portals. There will be a minimum requirement for stubs, for example stubs for monthly reports, and stubs for publications, meetings, and conferences that we need evidence for.
5. Allow users to reference external objects using stubs, as and when needed, or as part of the minimum stub set.

What is a Stub?

A stub is a special type of information object that combines some of the features of other basic content types in our content management portals. The diagram below sheds some more light:



In brief:

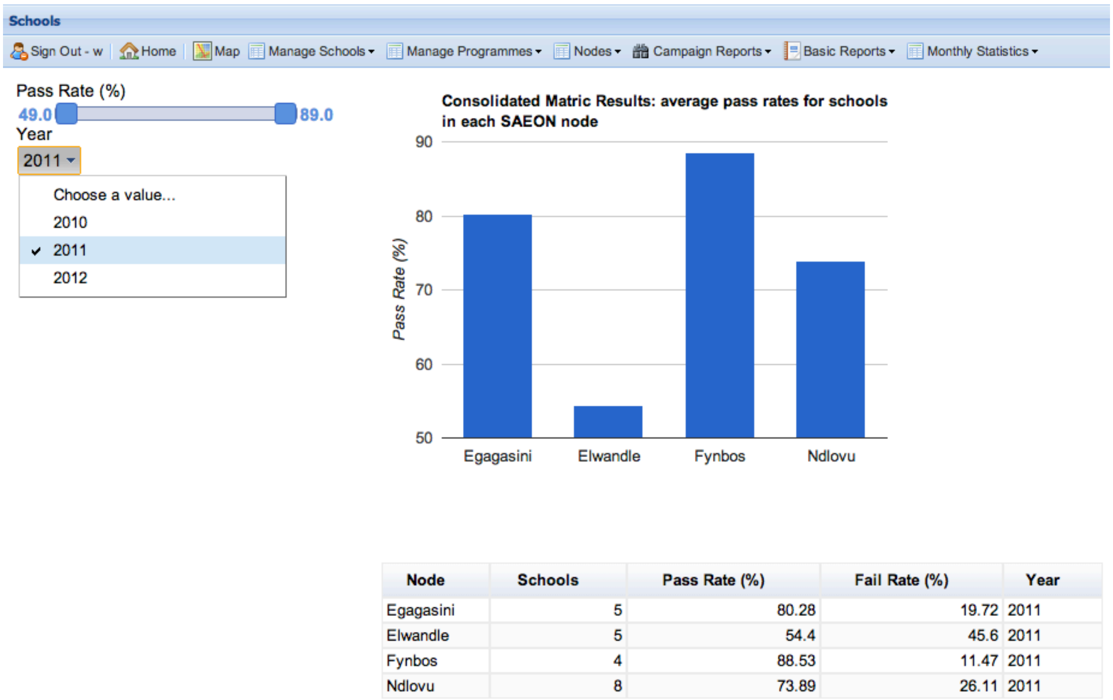
- Stubs have typical attributes such as a Title, an Abstract, and an Owner (within the portal).
- The Owner is sometimes the author, but not always.
- It has a publication date and a status, but this is not always the date to be used for 'reporting'. The effective of reporting date can be supplied separately.
- A stub can be tagged using standard keywords derived from our dimensions (Persons, Projects, Nodes, etc.)
- Users can add as many of their own tags as desired.
- Stubs can be linked to or provided with additional resources. These can be
    - An external URL;
    - An uploaded file;
    - An image;
    - An internal web page that can be anything from a list of hyperlinks to a full report. Any relevant material can be placed here.
- A Dublin Core meta-data record can be derived directly from the contents of the stub.

Reporting and Monitoring Across Systems and Sources

Our current portal and BigTable implementation provide standardized query services already. This allows us to query any part of the Knowledge Repository, Data Portal, Corporate Site, or BigTable either individually or in combination[2].

An example of such reporting is shown below:

[2] It is foreseen that a set of operational management and KPI reporting views will be developed over a period of 3 months – starting in April 2014.



Pass Rate (%) 49.0 89.0

Year 2011

Node	Schools	Pass Rate (%)	Fail Rate (%)	Year
Egagasini	5	80.28	19.72	2011
Elwandle	5	54.4	45.6	2011
Fynbos	4	88.53	11.47	2011
Ndlovu	8	73.89	26.11	2011

These reports are constructed on the fly, and hence are always up to date. It may be prudent to transfer data from the portal environments to BigTable databases on a periodic basis, since the portal content is transient and reporting data need to be preserved. This will probably be a monthly or quarterly transfer.

Tools are already available for report construction, and it is foreseen that key personnel at National Office will be able to add to and amend the collection of reports.

Annexures

A: Object Families

We have to provide for a number of different 'Object Families' in our Knowledge Repository, and accommodate them from systems both within and outside our control. Moreover, these systems vary from very formal (Financial System) to very informal (collections of unstructured data items stored in an online repository).

System Control	Name	Purpose	Family	Objects	Dimensions
NRF	Great Plains	Financial Management	Formal Relational Table Records	Budgets Asset Details	Cost Codes Asset Number
NRF	HR	Human Resources Management	Formal Relational Table Records	Personal Details	Persons
NRF	eRDMS	Records Management	Formal Relational Table Records, Objects	Documents	Persons Archive/ Filing System
NRF	RIMS	Grant Management	Formal Relational Table Records	Grant Data	People Institutions Grants
SAEON	BigTable[3]: People	SAEON Personnel, Collaborators, Students, Associates	Flexible Relational Table Records	Personal Details (additional to HR)	Persons
SAEON	BigTable: IT Assets	SAEON ICT Assets	Flexible Relational Table Records	ICT Asset Data (additional to Great Plains)	Asset Number
SAEON	BigTable: Schools	SAEON Education Outreach	Flexible Relational Table Records	Programme Data	KPI Node Project/ Programme
SAEON	Projects DB	SAEON Formal Projects	Formal Relational Table Records	Project Data	People Node Project KPI (Grant)
SAEON	BigTable: Projects	SAEON Informal Projects	Flexible Relational Table Records	Project Data (additional to Projects DB).	People Node Project KPI Grant
SAEON	BigTable: Grants	SAEON Grants	Flexible Relational Table Records	Grant Data (additional to RIMS)	Grant People Project

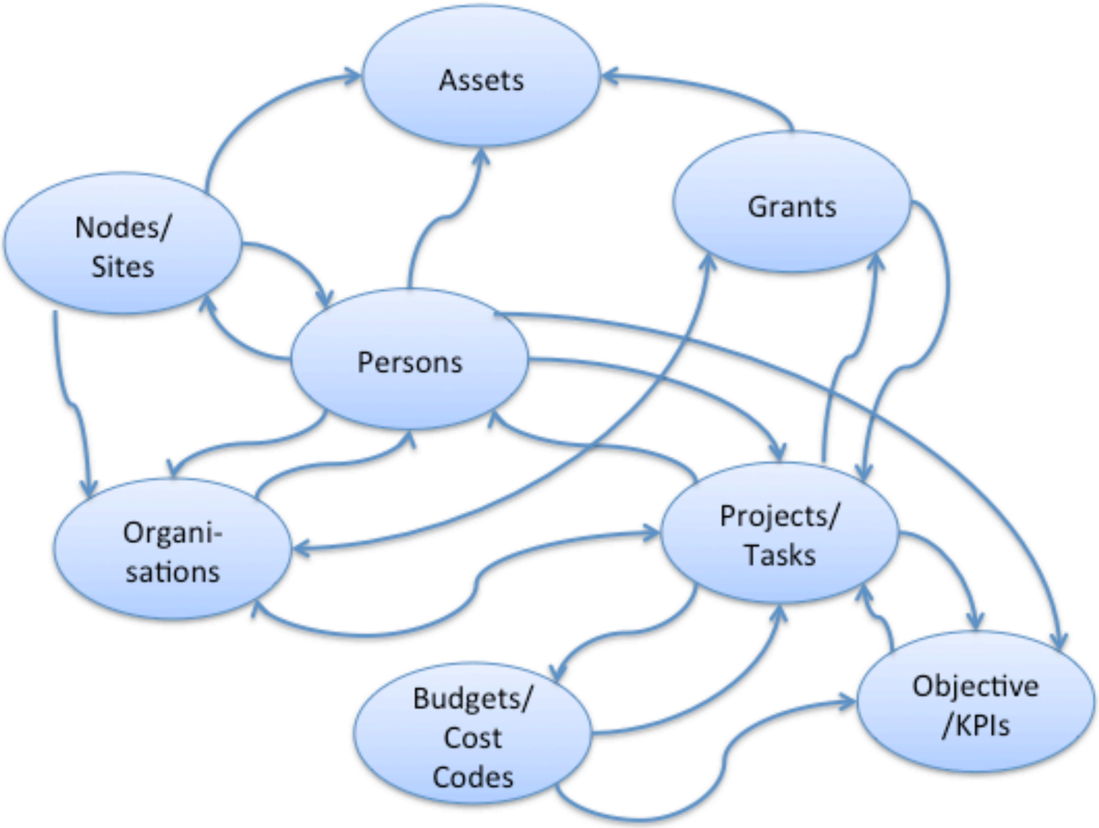
In terms of less formal objects, we need to accommodate the following:

[3] **BigTable**: A general purpose relational database that can be extended by users to accommodate a variety of small systems.

System Control	Name	Purpose	Family	Objects	Dimensions
SAEON	FileServers Shared Drives	Unspecified	File System/ Documents	Any	Any
SAEON	SAEON Data Portal	Data Storage	File System/ Documents/ Web Folders	Meta-Data/ Data Objects	Location People Institutions Multiple Tags
SAEON	SAEON Corporate Site	Communications News	File System/ Documents/ Web Folders	News Items Documents	Location People Institutions Nodes Projects KPIs Multiple Tags

B: Corporate Dimensions and a Conceptual Model

Most organisations have a fairly predictable core set of (information) ***dimensions***: these relate to the people, offices/ sites, objectives, tasks and projects, assets and other resources, external relationships, budgets, and outputs that are associated with the organisation. SAEON is no exception, and these dimensions should provide a large part of the formal 'tagging' (filing, classifications, or annotations) that we perform for the objects that we manage.



Many of these dimensions are, or should be, managed in more formal systems. Examples include the HR system, a financial system, or a projects management system. Needless to say, one has to accommodate the fact that these lists may not be exhaustive: as an example, not all the people that we wish to view as 'SAEON Personnel' will always be on our payroll, and we have many variations of people participating in our activities: students, research associates, collaborators, etc. It is clear that while we should not be using another source of data for the list of people in the HR system, but it is equally clear that it often needs to be supplemented in some way.

Because our tagging system is flexible, it is not a major issue if our initial model is not complete or has to be modified over time. We are also not too prescriptive about details: yes, persons have a recursive relationship that defines a reporting structure, and yes, projects and tasks and programmes form a hierarchy – but as long as we include one of the tags from a hierarchy branch when classifying an object, we can always find its other aggregate relationships outside of the knowledge network. It helps of the tag is fairly low down in the hierarchy, though.

The table below summarises our current thinking in this regard:

C: Interface Standards

To be completed.

D: An Emerging Knowledge Network

We will, in time, be able to construct a Knowledge Network from the objects that we gather and tag in the repository. This is achieved by assigning relationship values to the links between objects.

As an example, consider the situation where we have tagged a Conference Presentation with two 'Persons' associated with SAEON. Because of the context, we can qualify the relationship between the persons and call it something like 'Collaboration'.

This extension is some way into the future, but our design and underlying architecture allows us to mine our knowledge repository as required and construct a knowledge network (or more than one) onto it. These knowledge networks are essentially subsets or reductions of the maximum network created by all the links between all the objects.