15) PATRICK LAMBE & LOH YU GENE

Using a graph database to build a highly scalable and flexible knowledge mapping tool with taxonomy management system connectivity

1. About the Case Organization

Straits Knowledge is a small consulting firm that specialises in knowledge audits, KM strategy development and taxonomy development. Part of the knowledge audit methodology is a collaborative knowledge mapping activity.

2. About the Challenge

Our methodology for knowledge mapping for the past decade has been paper-based. We have a high burden of transcription of knowledge asset descriptions into large and complex Excel tables, checking for accuracy, and manual analysis of the knowledge maps. Analysis of the knowledge maps needs to be highly flexible because different organisations suggest different questions to explore.

An additional use for the knowledge maps is to provide raw evidence for the first draft of a faceted taxonomy for KM purposes (the maps represent a comprehensive description of key organisation activities and associated knowledge assets, in the language of the user community). Extracting this language from the excel spreadsheets has been laborious and burdensome.

Prior to the development of the graph-based system, which we call Aithin™, the work involved in transcribing, checking and analysing the maps was very expertise and labour-intensive. In addition, the maps were very difficult for the client to handle and analyse, and troublesome to update.

3. What We Did

We decided to go with a graph database solution for the following reasons:

- The maps needed to be highly scalable – we work with a wide range of organisations, from small to very large.
- Making the transition from a paper-based system tightly controlled by us to a system largely controlled by the client meant we needed to take an iterative, prototyping approach to development. Not all the issues and requirements could be anticipated in advance, and graph databases are intrinsically flexible in this way. The main thing we had to be clear about was to get the data model correct – after that, development could be very flexible.
- Making the application available to clients meant we could not predict all of the possible types of reports that might be needed. Graph databases, given a robust initial data model, are extremely versatile in the kinds of reports and analytics they can power.
- We needed to be able to offer project-based hosted licenses as well as permanent installed licenses – this meant the solution had to be browser-based, lightweight, and extremely fast.

The solution was innovative not only because this solution is unique in the market, but because it is leveraging the graph database in a number of different ways, including the capability to export data that can be consumed for taxonomy development.

4. Challenges and Lessons Learned

Most of our challenges related to being inexperienced in software development. We were very fortunate to be introduced to a development company that was familiar with knowledge management and taxonomy solutions, and so understood the purposes for which the software was to be deployed – and fortuitously, also had expertise in graph database development.

The flexibility of the graph database software also meant that we did not have to have all our requirements tightly tied down at the beginning, we could quite quickly overcome the usability and functionality issues we encountered along the way, simply because of our inexperience at the beginning.

While the cycle time from beginning of development to release of a working beta was longer than expected, much of this delay was caused by our need to develop helpware to guide our clients through the knowledge mapping methodology on the screen.

Our main advice would be to find a software development partner with a strong track record in developing graph solutions, and who understands, or is prepared to understand, the core tasks you are trying to perform.

5. Impact and Benefits

Key benefits include:

- Rapidly reducing the consulting time taken to collect maps, validate them and produce the analysis
- Enabling mapping activities to take place remotely (on the hosted version we can monitor and provide feedback on a mapping

session as it happens live, without having to be physically present)

- The data in the application can be rapidly analysed and is capable of being exploited in very flexible ways – the time taken to extract candidate terms for a taxonomy is also vastly reduced

6. Next Steps

Our first version was deliberately very simple, so as to make it accessible to clients using it for the first time. In our first handful of projects we have picked up a number of requirements to improve the usability and the interactivity of the analytics, and we are now working on version 2 with those features added. In the next version after that we plan to build a connector that will allow us to do direct export of map data into a taxonomy management system such as Synaptica®, to shorten and simplify the taxonomy development process.

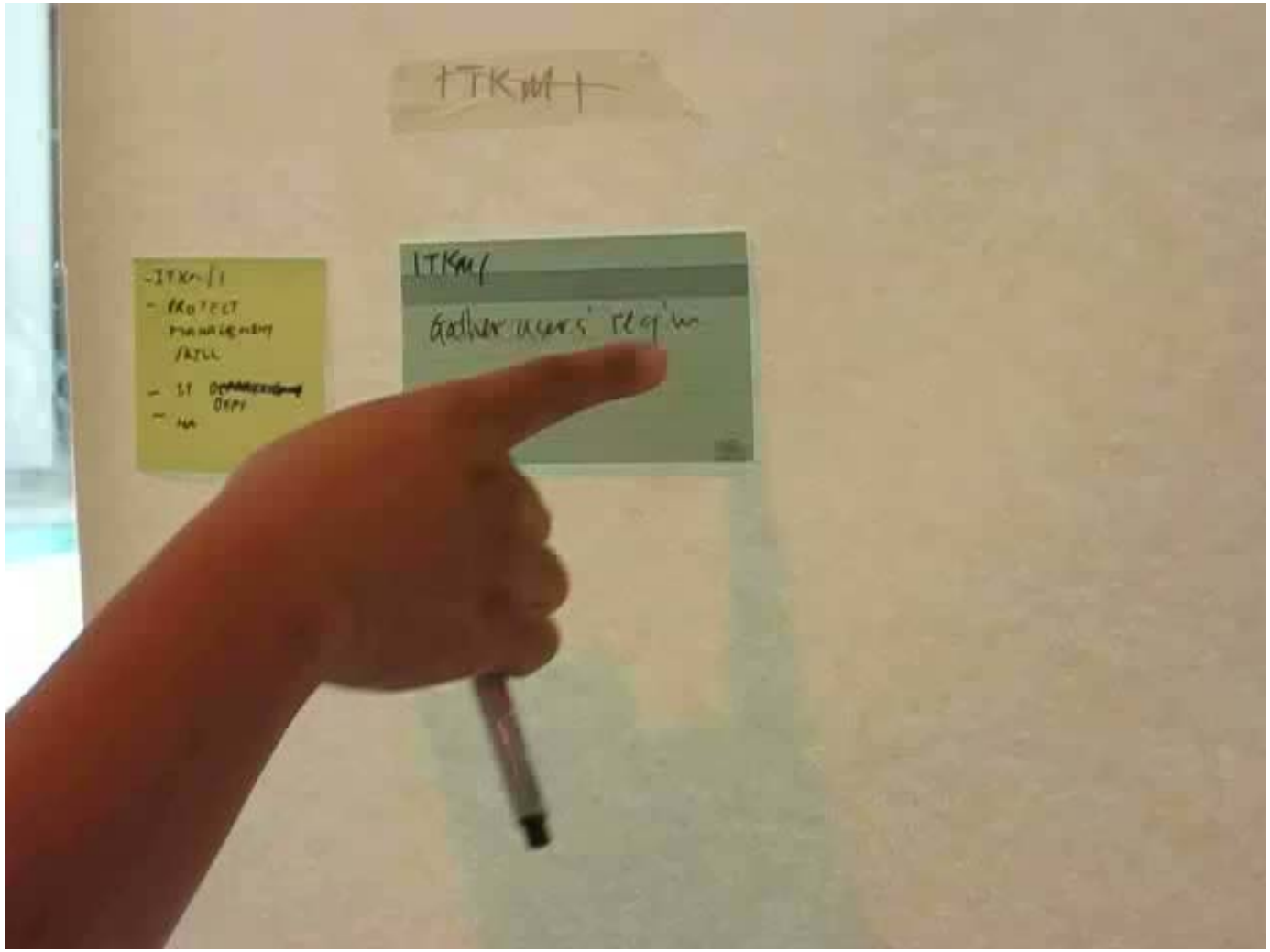
INNOVATIONS
IN KNOWLEDGE
ORGANISATION

Graph-based knowledge mapping tool with taxonomy management system capability

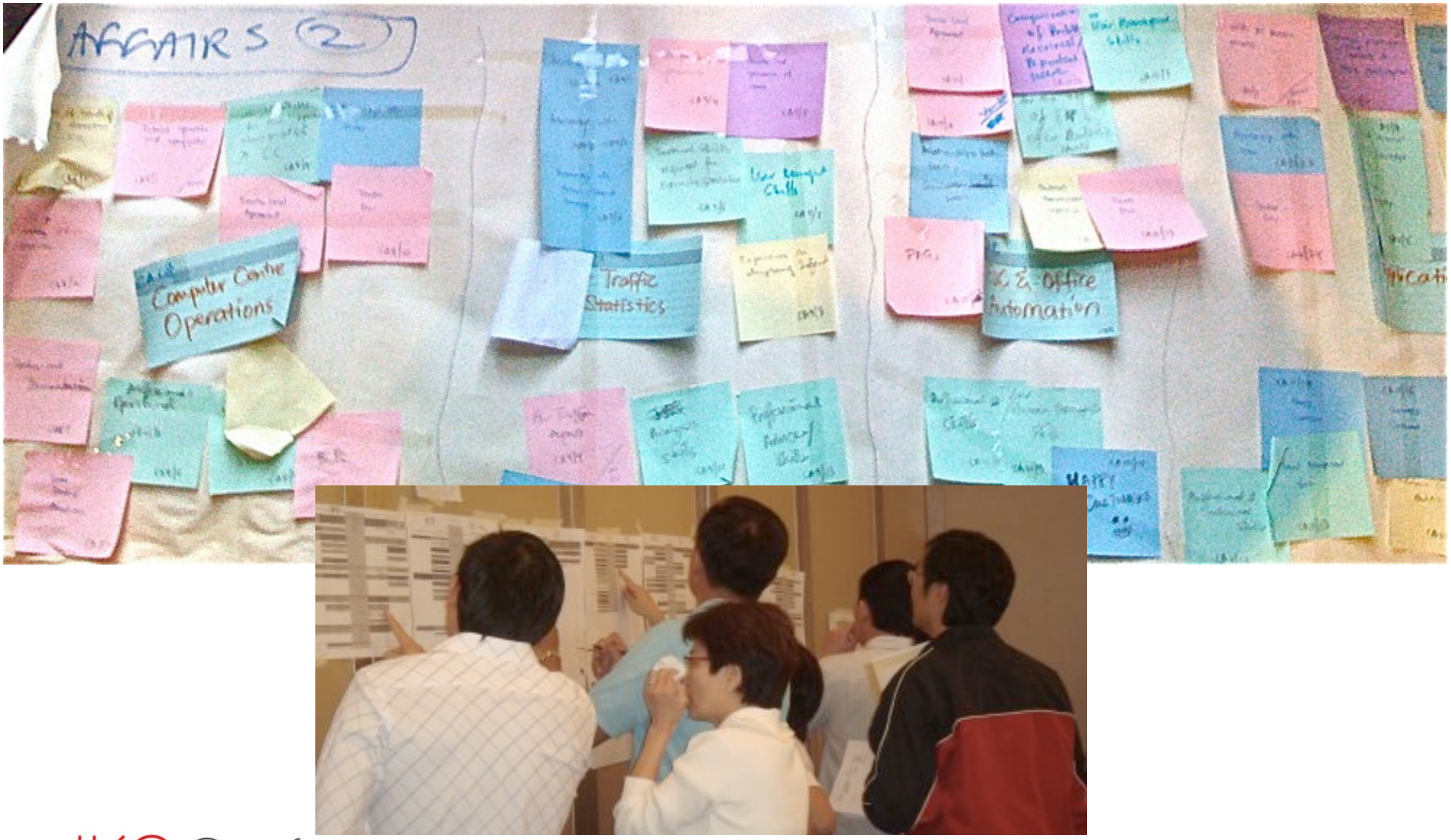
Patrick Lambe & Loh Yu Gene

#ikosg2015

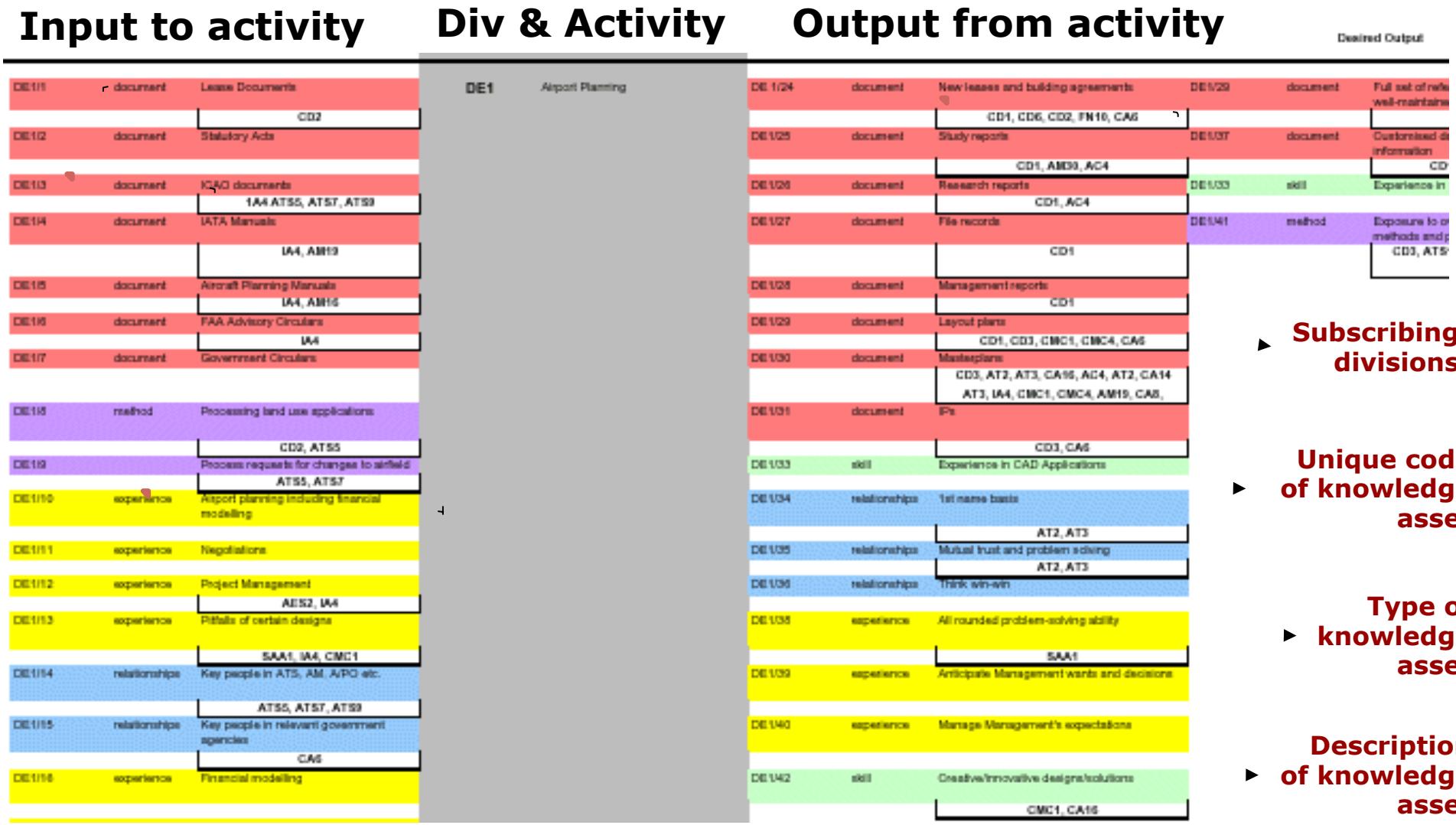
Knowledge Audits



Knowledge Audits



Knowledge Audits



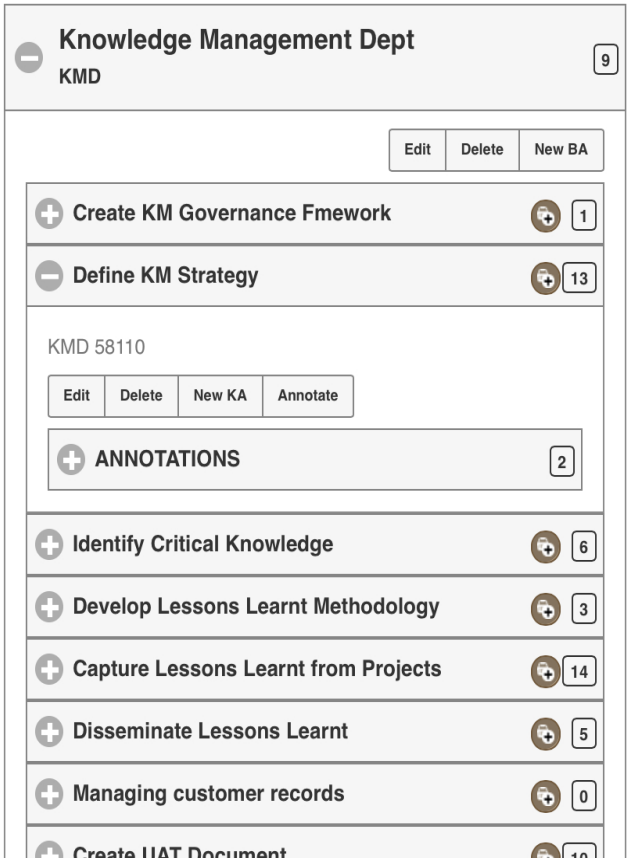
Moving to Digital

Moving to Digital

Aithin Knowledge Mapping & Analytics

Browse & Search

Workgroup/Activities

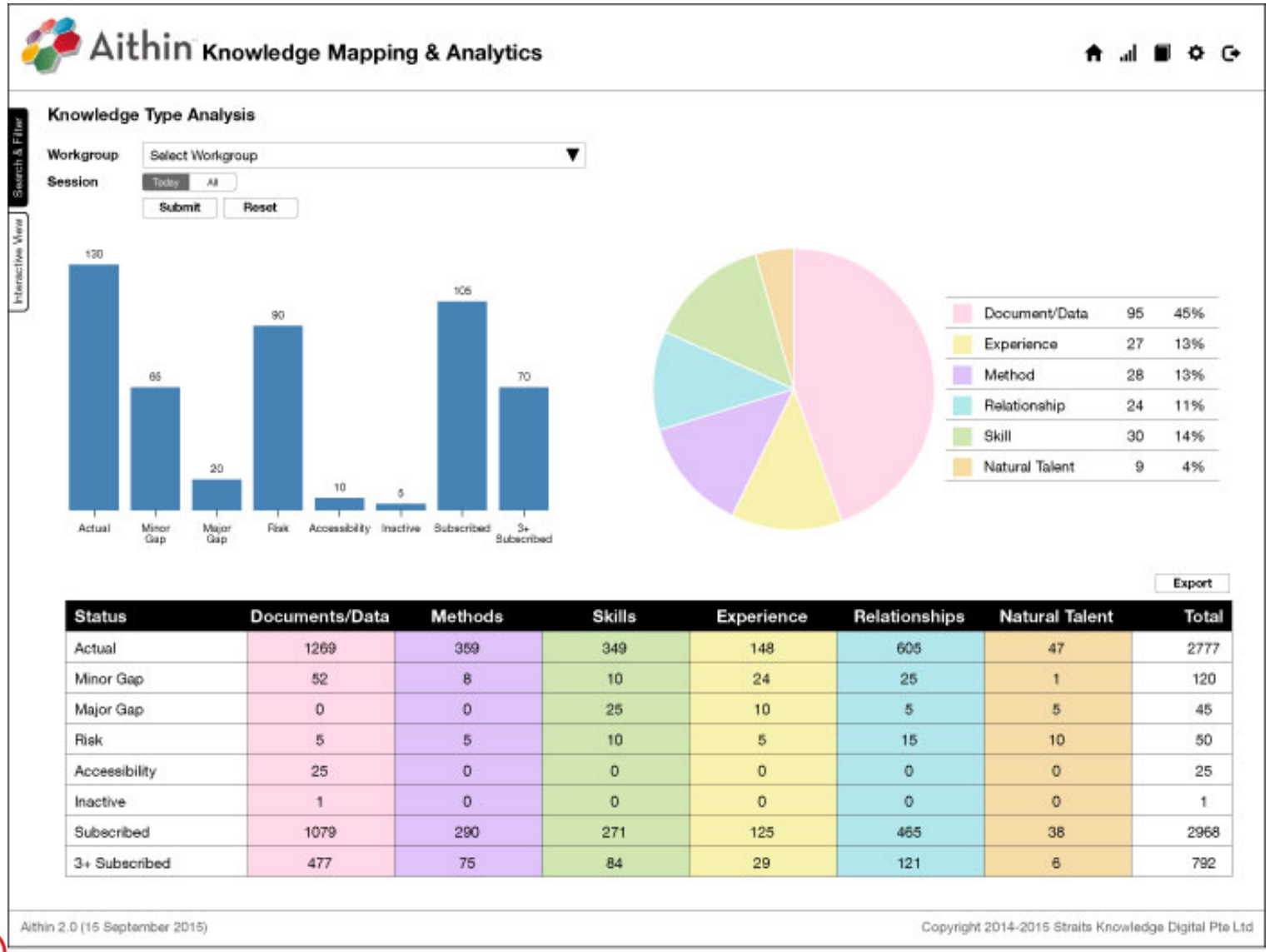


Knowledge Assets

Why Graph?

- Scalability – very small to very large organisations
- Flexibility – unpredictability of move from paper to digital interface
- Reporting and analytics
- Different licensing models – fast, lightweight, browser based
- Exploitation of maps for taxonomy development

What's Next?

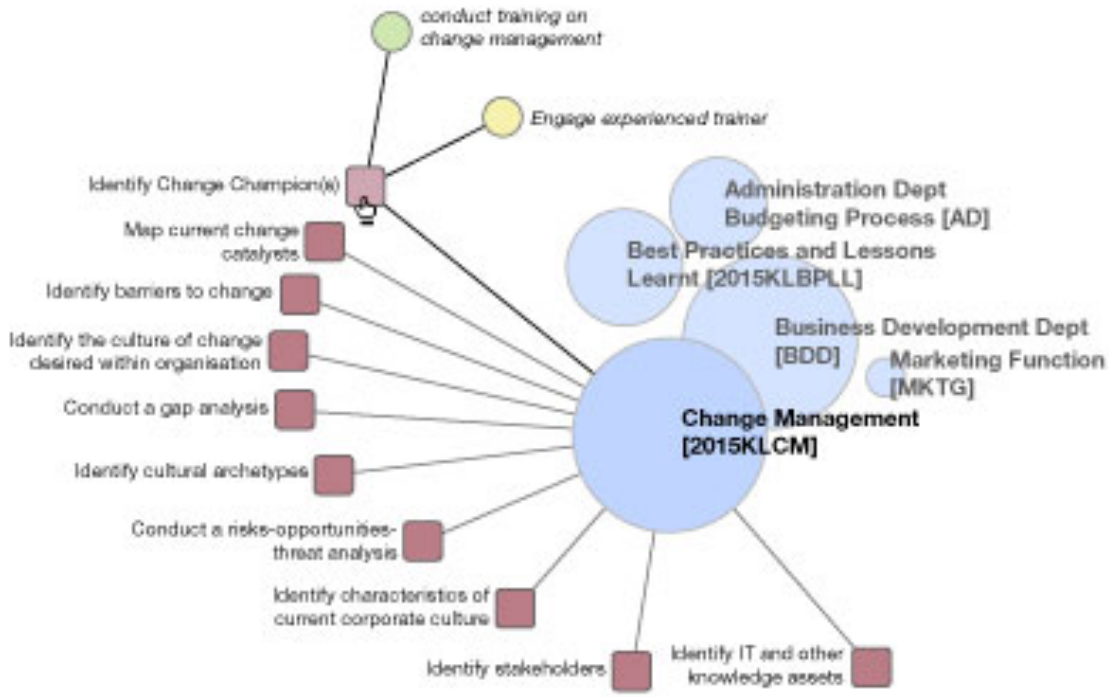


Status	Documents/Data	Methods	Skills	Experience	Relationships	Natural Talent	Total
Actual	1269	359	349	148	605	47	2777
Minor Gap	52	8	10	24	25	1	120
Major Gap	0	0	25	10	5	5	45
Risk	5	5	10	5	15	10	50
Accessibility	25	0	0	0	0	0	25
Inactive	1	0	0	0	0	0	1
Subscribed	1079	290	271	125	465	38	2968
3+ Subscribed	477	75	84	29	121	6	792

Search & Filter

Interactive View

Knowledge Map - Interactive View



Selected "Business Activity"

Export

KA Code	BA Name	Asset	Type	Input/Output	Status	Subscribed by	Activity Count	WG Count
2015KLCM 58815/A58849	Identify Change Champion (s)	conduct training on change management	Skill	Input	Minor Gap	-	0	0
2015KLCM 58815/A58853	Identify Change Champion (s)	Engage experienced trainer	Experience	Input	Minor Gap	FD 58099	1	1

Showing 1 to 8 of 9 rows 8 records per page

« ‹ 1 2 › »

Come and find out more at Topic #15!

plambe@straitsknowledge.com

Using a graph database to build a highly scalable and flexible knowledge mapping tool with taxonomy management system connectivity

Technical Addendum

LYG 20150527

Challenges and Lessons Learned

While the graph database platform chosen is flexible, it is also constantly updated, which causes backward compatibility issues that may need to be addressed in our application logic from time to time. In particular, we have seen the graph database query language, Cypher, evolve during the course of developing the first version of Aithin.

While these rapid changes are in contrast to a standards-based language like SPARQL, we see the benefits of a query language that is more powerful and more tightly integrated to the underlying database.

It was very clear to us from the beginning that a graph database is well suited to the methodologies employed by Straits Knowledge, since business logic and workflows are effectively represented in graph models. The technical challenge was to identify a suitable model-view-controller (MVC) application framework that fits our development environment.

Traditional SQL databases often have an abstraction layer embedded into the *model* component of the MVC, which is shielded from the *controller*, so that an application developer does not necessarily have to interact with the database.

We decided to use a basic MVC structure, and adapt it by building plugin modules to abstract the *model*.

Impact and Benefits

A benefit of using graph database is the ability to extend our data model with little or no change to application logic, resulting in greater source code reusability as new features are being developed for Aithin.

Additionally, as we update the data model and add functionality, application complexity remains relatively linear, making it easier to scale up.