AN EVENT-DRIVEN CARTOGRAPHIC APPROACH TO MODELLING SOFTWARE ENGINEERING KNOWLEDGE

Ernest Cachia and Mark Micallef
Department of Computer Science, Faculty of ICT, University of Malta, Msida, Malta

Keywords: Knowledge maps, Knowledge mobility risk, Knowledge transfer risk, Software engineering, Knowledge metrics.

Abstract: This paper proposes an event-based cartographic approach to managing software engineering knowledge that goes beyond the typical yellow-pages paradigm. After proposing a way of modelling knowledge assets, persons in an organisation and various relationships between these elements as a graph, the authors go on to demonstrate how this approach can be useful. Since the model is represented by a mathematical structure, established techniques from graph theory can be used for interesting analysis such as detecting knowledge risks, modelling staff turnover scenarios and identifying people who have similar knowledge to each other. The proposed approach uses an event-driven paradigm which infers the strength of knowledge relationships based on individuals' participation in each knowledge asset's life cycle.

1 INTRODUCTION

Software engineering is a knowledge-intensive activity. For software organisations, the main assets are not manufacturing plants, buildings, and machines, but the knowledge held by the employees (Bjørnson and Dingsøyr, 2008). Software engineers are not merely vessels of technical knowledge but are fully fledged knowledge workers. They are expected to form a deep enough understanding of whatever domain they happen to be working in such that they are able to apply their technical knowledge to build solutions which solve problems in that domain. In an industry with high staff turnover rates, this can be worrying. Organisations need to ensure that as employees flow in and out of the their employ, the intellectual capital they create and work with is somehow retained and leveraged to increase their competitive edge. Studies (Linberg, 1999)(Pressman, 1998) have shown that projects do not tend to fail because of developers' lack of technical knowledge, but rather for reasons such as *requirements failures*, *communication failures* and *estimation failures*. These failures can be traced back to inadequate knowledge management practices as a root cause.

Earl's classification of knowledge management schools (Earl, 2001) is widely cited in the literature and classifies knowledge management schools into three broad categories: *technocratic*, *behavioural* and *economic*. The work presented here involves a school of thought within the technocratic category: the *cartographic* school. The driving principle here is that of connectivity. That is to say, maximising the use of knowledge within the organisation by focusing on leading knowledge seekers directly to the knowledge providers who can satisfy their needs. This is usually accomplished using a yellow-pages style directory and has been shown to be useful for allocating resources, searching for competence, identifying project opportunities and upgrading skills (Dingsøyr et al., 2005) (Bjørnson and Dingsøyr, 2008).

In this paper, it is argued that the cartographic school lends itself to the dynamic nature of software engineering organisations. A more elaborate cartographic approach is proposed which will not only lead knowledge seekers directly to knowledge providers but also provide organisations with the ability to model 'how much' each person knows a particular knowledge asset therefore enabling them to detect knowledge mobility risks, knowledge transfer risks, model staff turnover scenarios and also analyse various structural characteristics of their organisational knowledge (see section 4).

In the following sections an overview of the modelling approach is given, followed by an exploration of the analysis capabilities that the approach provides.

Cachia E. and Micallef M..
AN EVENT-DRIVEN CARTOGRAPHIC APPROACH TO MODELLING SOFTWARE ENGINEERING KNOWLEDGE.
DOI: 10.5220/0003630400180027
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2011), pages 18-27
ISBN: 978-989-8425-81-2

1.1 A Note on Scope

It is worth noting that the work presented here has been formalised into a formal language with appropriate syntax and semantics (Micallef, 2011). However, it is beyond the scope of this paper to give a detailed mathematical analysis of the technique. Rather, the aim is to give a practical overview of the technique and its uses for critique by the knowledge management community. Also, although the work here is framed within the context of software engineering, most of the concepts presented can be applied to any knowledge-intensive industry. The main reason for this work being linked to software engineering is to provide a cohesive link with other research that the authors are carrying out.

2 PROPOSED APPROACH

The proposed approach seeks to allow organisations to build a model of their organisational knowledge and represent it as a graph. In this graph, each vertex represents either a knowledge asset or a person. It is important to note that a vertex representing a knowledge asset does not necessarily imply that the knowledge asset is known by the organisation, but rather that it is recognised as being of value to it. Knowledge assets are discussed in more detail in section 2.1 whilst persons and teams are the subject of section 2.2. Edges in the graph represent relationships between two knowledge assets or relationships between a person and a knowledge asset. These are discussed in section 2.3.

This work also proposes the modelling of events. Events represent various actions by persons with respect to knowledge assets and are used to infer who knows what within the organisation. The main idea here is that the more a person participates in the life cycle of a particular asset, the more familiar they are likely to be with it. A mechanism for modelling the deterioration of knowledge is also presented. Events are discussed in detail in section 3.

2.1 Knowledge Assets

Knowledge assets represent knowledge which may or may not be known by persons in the organisation but has been deemed to be of value to it. Vertices which represent knowledge assets have a number of properties associated with them. These are as follows:

Name - a unique identifier for the knowledge asset.

Category - categorises the knowledge asset as being *technical*, *business* or *general*. These categories are an adaptation of the ones proposed by (Ramal et al., 2002) who proposed that software engineers know three categories of knowledge: *computer science*, *business* and *general*. The authors felt that the renaming of the “Computer Science” category to “Technical” was required because the term “Technical” knowledge provides an umbrella term for knowledge which may have otherwise been confusing given that computer science refers to a specific subset of topics in the academic world.

Visibility - refers to the classification of the knowledge asset as *tacit* or *explicit*. This classification of knowledge is widely cited (Alavi and Leidner, 2001)(Duffy, 1999)(Tiwana, 2000) and divides knowledge based on whether it resides purely within its ‘knower’ (tacit knowledge) or whether it has been explicitly articulated, codified or otherwise communicated (explicit knowledge). The motivation for including this classification in the model is that a knowledge asset’s visibility has an impact on the retainability and transferability of particular knowledge assets. Szulanski points out that tacit, context-specific and ambiguous knowledge is likely the most difficult to transfer within the firm (Szulanski, 1996). If an organisation is able to identify tacit knowledge in its knowledge map, it would be able to identify potential problem areas that would occur if for example key people who know critical tacit knowledge were to leave the organisation.

Social Classification - classifies a knowledge asset as being *individual* or *social* and was proposed by (Nonaka, 1994). Individual knowledge is knowledge that is created and homed within an individual. On the other hand, group knowledge is created and inherent in the collective actions of a group, with no individual member possessing all the knowledge. This property is included in the model for two reasons. Firstly it provides a means for statistical analysis of organisational knowledge from the social point of view and secondly, the balance between individual and social knowledge has an implication on the ease with which that knowledge is shared amongst specific individuals.

Operational Classification - classifies a knowledge asset as declarative (know-about), procedural (know-how), causal (know-why), conditional (know-if) or relational (know-with) (Nolan Norton Institute, 1998)(Zack, 1998). This property is included because if provides an interesting operational perspective on the different types of knowledge that an organisation deals with. Making this property visible may lead to a situation whereby management realise that there is a potentially harmful imbalance in the types of knowledge that its employees know (e.g. too much know-how as opposed to know-about).

It is worth noting that all properties discussed here except for the *name* property can be set to *unkown* if the organisation does not feel it is beneficial to model a particular property about its knowledge assets.

2.2 Persons and Teams

Person vertices represent individuals within the organisation and exist mainly for the reason of modelling knowledge relationships between individuals and knowledge assets (see section 2.3). Consequently, a person has only two properties. The first is a *name* which acts as a unique identifier for the person. The second is a list of *teams* which the person is a member of. Teams are sets of persons clustered together for some scope determined by the organisation. From the knowledge map point of view, teams are important because they enable reasoning about groups of people.

2.3 Relationships

Relationships constitute the main feature of the proposed approach that provides an advantage over traditional yellow-pages style cartographic approaches. They serve two main purposes. Firstly, they provide visibility into the relationship between individual knowledge assets. This allows organisations to reason about structural properties of their knowledge landscape and also carry out more efficient knowledge transfer exercises, skills analysis, interview design, and so on. Secondly, when used to link a person to a knowledge asset, they provide an indication of who knows what and 'how much' they know it. Four types relationships are defined:

Related Relationships - A *related* relationship is a non-directed edge between two knowledge assets and signifies that the two knowledge assets are related in some way. This relationship is the weakest form of relationship between two knowledge assets in the proposed language.

Dependency Relationships - A dependency is a directed edge between two knowledge assets and signifies that one knowledge asset depends on another. That is to say, in a scenario where a knowledge asset k_1 depends on another knowledge asset k_2, if a person p is to utilise or learn k_1, she must first know k_2. A typical example of a dependency relationship would be "Java depends on OOP" since it is unlikely that someone can be an effective Java programmer unless they have a firm grasp of the object oriented paradigm.

Composition Relationships - A composition is a directed edge between two knowledge assets and signifies that one knowledge asset forms part of another. That is to say that the latter is partially composed of the former. A typical example would be "JDBC is part of Java". Modelling this type of relationship allows organisations to structure their model in a more readable way but also allows for reasoning such as "if a knowledge asset k_1 is composed of two knowledge assets k_2 and k_3, if a person p knows k_2 then she also knows k_1 albeit to a lesser degree.

Knowledge Relationships - A knowledge edge is a directed edge that connects a *person* vertex to a *knowledge asset* vertex so as to signify that the person 'knows' the knowledge asset to some extent. This is type of relationship is typically not directly specified but is automatically inferred through the use of events (see section 3). Knowledge relationships also exhibit a *magnitude* property which is an indication of 'how much' the person knows the knowledge asset.

At this point, it is worth illustrating the work presented so far in the context of an example.

Example 1 . *Consider a particular scenario whereby an organisation employs a single team consisting of two people (Chris and Jane). The organisation has identified seven knowledge assets of value: [1] oop, [2] java (depends on oop), [3] sql, [4] web development, [5] smalltalk (depends on oop), [6] jdbc (part of java) (depends on sql), and finally [7] servlets (part of java) (related to web development).*

As mentioned in section 1.1, the authors have developed a formal language for building such models and have also created an english-like domain specific language which provides syntactic sugaring for the formal language (Micallef, 2011). It is beyond the scope of this paper to present the language but the scenario presented in example 1 would be represented as follows:

```
team developers;
person Chris member of developers;
person Jane member of developers;

knowledge asset oop
category technical;

knowledge asset java
category technical
sociality social
depends on oop;
```

```
knowledge asset sql
sociality individual
operationality procedural;

knowledge asset web_development ;

knowledge asset smalltalk
depends on oop;

knowledge asset jdbc
depends on sql
part of java;

knowledge asset servlets
part of java
related to web_development;
```

For the sake of brevity, example 1 does not completely specify all the knowledge assets' properties but gives examples of each property being used in at least one knowledge asset. Figure 1 provides a graphical representation of the model. One can notice that *related* relationships are represented by a dashed edge, *dependency* relationships are represented by a continuous edge with a solid arrow head, and *composition* relationships are represented by a continuous edge with a hollow arrow head.

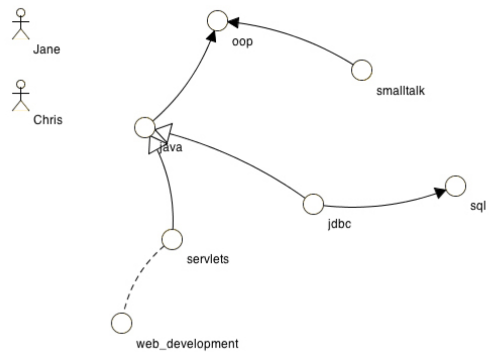

Figure 1: Graphical depiction of the model created by example. 1

Figure 1 also indicates that thus far, neither Jane nor Chris actually *know* any of the knowledge assets which the organisation considers valuable. Information about knowledge relationships will be inferred from events as discussed in section 3.

3 EVENTS

Events provide the ability to model and reason about knowledge relationships between persons and knowledge assets. A mechanism is proposed whereby various interactions between persons and knowledge assets are logged and analysed to infer the existence and magnitude of knowledge relationships. Three categories of events are defined: knowledge events, resource events and time events. The following sections will look at each event category in turn.

3.1 Knowledge Events

Knowledge events model a knowledge asset's journey through various stages of its life cycle as well as the persons involved in each transition. The following knowledge events are defined:

Knowledge Identified. Refers to an event whereby a knowledge asset is identified as being of value to the organisation and thus starts being tracked.

Knowledge Created. This event occurs when new knowledge has been created within the organisation. This could involve someone reading a book, attending a training course, or even codifying her own tacit knowledge into an explicit form.

Knowledge Stored. Models a situation where explicit knowledge is stored in some way, shape or form.

Knowledge Retrieved. Refers to the act of someone retrieving explicit knowledge from storage.

Knowledge Transferred. This is one of the most important events in a knowledge organisation and refers to the transfer of knowledge from one or more knowers to one or more learners.

Knowledge Modified. Indicates that a particular knowledge asset has been modified in some way. This event is useful for the purposes of notifying anyone connected with the knowledge asset that it has changed.

Knowledge Applied. This event models a period of time during which one or more persons are making use of a particular knowledge asset so as to achieve some value for the organisation.

Knowledge Discarded. Indicates that a particular knowledge asset is no longer of any value to the organisation.

All knowledge events have a date associated with them and most have a set of persons involved in the event. However, *Knowledge Identified* events do not have any associate persons whilst *Knowledge Transferred* events split the set of persons into two: the *knowers* who are transferring the knowledge and the *learners* who are on the receiving end of the transfer.

3.2 Resource Events

Resource events model the flow of people through the organisation. This is useful for a number of reasons. Firstly, a person leaving the organisation always results in the organisational knowledge landscape changing. Therefore it is desirable to keep our model up to date in this regard. Secondly, it is sometimes useful to reason about knowledge at the *team* level of abstraction. For example, one might ask the question "how would a team be affected if a particular member leaves the team to work with another team?" or "what knowledge transfer opportunities can occur if we move certain people around?". Finally, if hypothetical events are logged, an organisation can model possible future knowledge scenarios and take action if any risks are detected. For example, one could model an event where person p is leaving the organisation even thought he is not. In so doing, the organisation can get a view of what changes in the knowledge landscape could occur as a result of such an event and take risk-mitigating action if necessary.

The following resource events are defined:

Person Left Organisation. Refers to an event whereby a particular person leaves the organisation. This has repercussions on the organisations global knowledge landscape.

Person Left Team. Models a situation whereby a person remains employed within an organisation but is no longer part of a particular team. In this case, the organisation's knowledge landscape changes in the context of the team's knowledge.

Person Joined Team. This event models a situation whereby a person has joined a team within the organisation. Team membership is not exclusive so persons can potentially form part of multiple teams.

One might notice the ominous exclusion of a *Person Joined Organisation* event. Such an event would indeed be useful but its semantics would be such that it can be modelled as a sequence of knowledge events. That is to say, a person joining the company can be modelled by modelling the person's knowledge activities prior to joining up. Simply creating a *Person Joined Organisation* event will not achieve the desired effect on the model.

3.3 Time Events

Time events indicate the passage of time and thus allow us to construct a mechanism whereby we can model the deterioration of personal and organisational knowledge. The driving principle here is that if a person p does not participate in a knowledge asset k's life cycle for a certain amount of time t, then as t increases, p knows k less to some increasing degree. A single time event is defined:

Time Passed - Indicates the passage of a single time unit. It is recommended that a time unit be considered to represent a day but any time unit could be utilised according to the organisation's needs.

3.4 Effects of events on models

Events can also be categorised according to how they affect knowledge relationships in a model:

Strengthening Events result in the creation of new knowledge relationships or the strengthening of existing ones. For example, if a person p knows nothing about Java, then the model will not contain a knowledge relationship between the p and Java. However, if p attends a relevant training course, this can be logged as a *Knowledge Created* event which results in a knowledge relationship with some initial magnitude being created between p and Java. Furthermore, if p applies Java to her work on an ongoing basis, then the magnitude of the knowledge relationship increases over time as a result of *Knowledge Applied* events. This category of events includes all knowledge events except for *Knowledge Identified*, *Knowledge Modified* and *Knowledge Discarded* events.

Weakening Events result in the weakening or removal of existing knowledge relationships. Time events for example result in the weakening of all knowledge relationships which have not been subjected to any strengthening events in the most recent time unit. This category of events includes *Knowledge Discarded*, *Person Left Organisation*, and *Time Passed* events.

Agnostic Events have no result of any knowledge relationships because they exist for other reasons. For example, *Person Left Team* events do not affect a person's knowledge relationships in any way, even if at the *team* level of abstraction one can say that the team's knowledge has been weakened. However, vertices on the graph only represent persons with teams being only a conceptual entity. This category is composed of *Knowledge Identified*, *Person Left Team* and *Person Joined Team* events.

Hybrid Events have the potential of strengthening certain events whilst weakening others. For example, a *Knowledge Modified* event would strengthen the knowledge relationship between the person involved in the event and the knowledge asset but would weaken the relationship of others who are not involved in the event. The main reason behind this is that if a knowledge asset has changed then other people's knowledge of it may have become obsolete to some degree. The *Knowledge Modified* event is the sole member of this category.

Example 2. *Following on from example 1, consider a situation whereby the following events occur in sequence:*

1. *Both Jane and Chris take a course. Chris learns about Java whilst Jane learns about Smalltalk*
2. *Chris uses Java for one month whilst Jane uses Smalltalk for two weeks*
3. *Chris teaches Jane about Java*

The first event is represented as follows:

```
knowledge created java
by Chris on "2011-03-01";

knowledge created smalltalk
by Jane on "2011-03-01";
```

This leads to a knowledge model as depicted in figure 2. One can notice that there is now a knowledge relationship between Chris and Java of magnitude 1 (as indicated by the labelled edge) as well as a knowledge relationship linking Jane to Smalltalk, also of magnitude 1. The calculation of knowledge relationship magnitudes is explained in section 3.5. However, it suffices to say that the higher a the magnitude of a knowledge relationship, the more confident an organisation can be that the person in the relationship knows the knowledge asset.

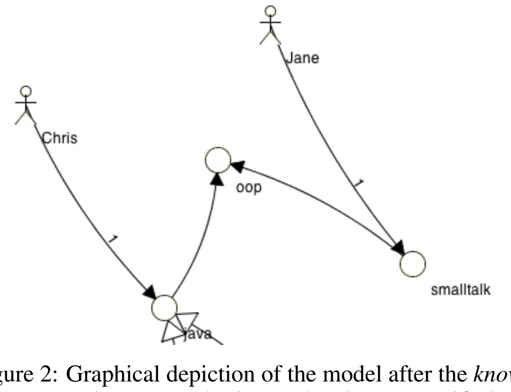

Figure 2: Graphical depiction of the model after the *knowledge created* events. The image has been magnified and cropped for clarity.

The next event in this example involves both Jane and Chris applying their knowledge of Smalltalk and Java for a period of two weeks and one month respectively:

```
knowledge applied java by Chris
from "2011-03-01" to "2011-03-31";

knowledge applied smalltalk by Jane
from "2011-03-01" to "2011-03-15";
```

Figure 3 indicates that the same knowledge relationships from figure 2 remain but their magnitudes have increased. Also, the magnitude of the knowledge relationship linking Chris to Java has increased by twice as much as the knowledge relationship linking Jane to Smalltalk. This is because the fact that Chris applied Java knowledge longer than Jane applied Smalltalk knowledge leads us to reason that we are more confident in Chris knowing Java than in Jane knowing Smalltalk.

Finally, we model the scenario where Chris transfers his knowledge of Java to Jane:

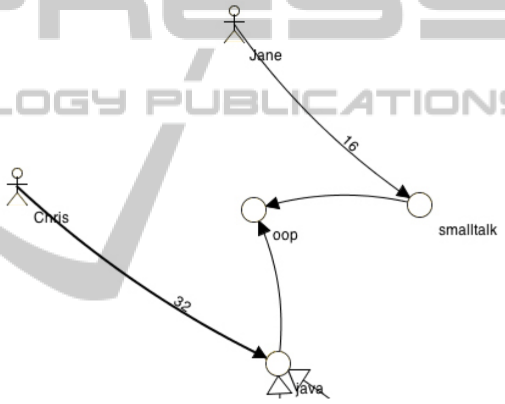

Figure 3: Graphical depiction of the model after the *knowledge applied* events. The image has been magnified and cropped for clarity.

```
knowledge transferred java
by Chris to Jane
on "2011-04-01";
```

As shown in figure 4, two things have happened. Firstly, a new knowledge connection has been created that links Jane to Java. This indicates that Jane now knows Java but knows it to a much lesser degree than Chris, who has been using the language for a month. Secondly, the magnitude of the knowledge connection linking Chris to Java has increased because the act of one person transferring knowledge to another actually strengthens the former's knowledge.

3.5 Knowledge Event Magnitudes

The amount by which a particular knowledge event strengthens or weakens the magnitude of a knowledge relationship will affect the extent to which the

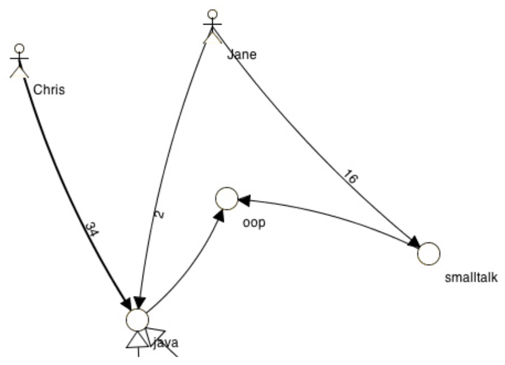

Figure 4: Graphical depiction of the model after the *knowledge transferred* events. The image has been magnified and cropped for clarity.

resulting model represents the true knowledge landscape of the organisation. Since realistic modelling will require substantial research, for the time being the authors propose to refer to an oracle when adjusting knowledge relationship magnitudes. It is very likely that a completely realistic mechanism will be very difficult to construct but the level of absolute accuracy is not the be-all and end-all of the work presented here. Rather than focus on the accuracy of 'how much' a person knows a knowledge asset, we are more interested in the relativity of the knowledge relationship magnitude when compared to other knowledge relationships. Given that all knowledge relationship magnitudes are calculated in the same way, this should minimise distortion and make the models fit-for-purpose when it comes to their analytical capabilities.

4 MODEL ANALYSIS

In this section the analytical properties of the proposed approach are discussed.

4.1 Inherent Model Properties

The model itself has a number of inherent properties which enable reasoning about the knowledge landscape in an organisation. The following is a list of the more salient properties:

4.1.1 Structural Properties of Knowledge Assets

Knowledge asset structural properties refer to properties that enable reasoning about the structure of knowledge assets in the model.

Symmetry of *Related* Relationships. Given two knowledge assets k_a and k_b, if k_a is related to k_b then k_b is related to k_a.

Transitivity of Dependencies. Given three knowledge assets k_a, k_b and k_c, if k_a depends on k_b and k_b depends on k_c, then k_a depends on k_c.

Transitivity of Composition. Given three knowledge assets k_a, k_b and k_c, if k_a is part of k_b and k_b if part of k_c, then k_a is part of k_c.

Composition as Dependency. Given two knowledge assets k_a and k_b, if k_a is part of k_b then k_a also depends on k_b.

Non-symmetry of Dependency. Given two knowledge assets k_a and k_b, if k_a depends on k_b then k_b cannot depend on k_a.

Non-symmetry of Composition. Given two knowledge assets k_a and k_b, if k_a is part of k_b then k_b cannot be part of k_a.

4.1.2 Properties of Knowledge Relationships

The following properties hold for knowledge relationships in any model constructed using the technique proposed in this paper.

Team Knowledge. If a person p knows a knowledge asset k and is a member of team t, then it holds that t knows k.

Transitivity of Knowledge through Composition. Given two knowledge assets k_a and k_b such that k_a is part of k_b, if a person p knows k_a then p also knows k_b to some degree.

4.2 Detecting Knowledge Mobility Risks

The term *knowledge mobility risk* refers to a chance of the company loosing a valued knowledge asset as a result of a particular person leaving. In general, the greater the number of people that people know a knowledge asset, the less mobility risk that knowledge asset exhibits. However, one must consider 'how much' each person knows the knowledge asset. If (for example) four people in a team are vaguely familiar with a critical part of a system but a fifth person is an expert about it, then with respect to that particular knowledge asset, loosing that one expert will probably hurt more than loosing any number of the other four members.

Since the proposed model is actually a graph, the authors looked to graph metrics for a way to calculate knowledge mobility risk. Of particular interest was work done by Botafogo which proposed *Relative Out Centrality (ROC)* and *Relative In Centrality (ROC)* as measures of the social importance of a vertex in a graph. Of particular relevance to knowledge mobility risk is the *RIC* metric, which is based on calculations involving inbound paths from other vertices into the vertex being analysed. However, since in the proposed model the maximum length of a knowledge path is one, Botafogo's approach was modified to take into account the magnitude of knowledge relationships (Micallef, 2011). This adapted *RIC* metric can be interpreted as a measure of knowledge mobility risk. The higher the RIC measure for a knowledge asset, the more knowledge mobility risk it exhibits.

Example 3. *The model depicted in figure 5 is the result of a three month evaluation exercise with a four person team. In order to analyse this particular model for knowledge mobility risks, the knowledge asset vertices are coloured according to the following criteria:*

1. *Let σ be the standard deviation of the RIC values for all knowledge asset vertices in the graph*
2. *Let μ be the average of the RIC values for all knowledge asset vertices in the graph*
3. *Colour all nodes with $RIC \leq \mu$ as green (represented as white in this paper)*
4. *Colour all nodes with $\mu > RIC \leq (\mu + \sigma)$ as orange (represented as light-grey in this paper)*
5. *Colour all nodes with $RIC > (\mu + \sigma)$ as red (represented as dark-grey in this paper)*

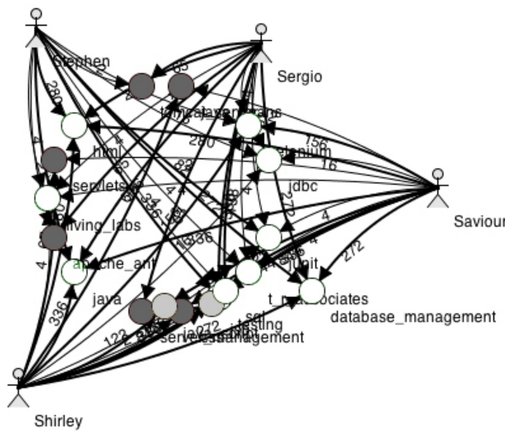

Figure 5: A knowledge model coloured according to knowledge mobility risk.

Figure 6 shows the same model filtered to show only red vertices. This enables one to analyse potential reasons behind the high knowledge risks and possibly take corrective action. In this particular example, the server_management *knowledge asset is only known by one person (Sergio). Furthermore, the fact that he seems to know it relatively well when compared to other scores in the model indicates that he does a lot of server management for the team. This is a knowledge mobility risk because if Sergio leaves there is no one in the organisation who can seamlessly take over his job. The other high-risk vertices are known by two people but in each case, one person knows the asset much more than the other, thus leading to a knowledge mobility risk. Having this visibility, the team may decide to mitigate the risk by (for example) having people pair together on high-risk assets until such a time when the risk is sufficiently reduced.*

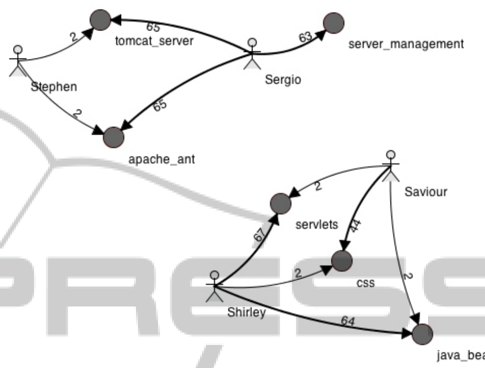

Figure 6: A filtered version of the model showing only high knowledge mobility risk assets.

4.2.1 Knowledge Mobility Risk from a Person Standpoint

Botafogo's work (Botafogo et al., 1992) also defines *Relative Out Centrality (ROC)* as a measure of a vertex's influence on other vertices in the graph. After again modifying this metric to take into account the magnitude of knowledge relationships (Micallef, 2011) applied ROC to inferring how knowledge mobility risk from a *person* point of view. That is to say, based on some subset of knowledge assets of interest, we can calculate if there are any persons who's knowledge of these assets is so strong that it causes an imbalance and subsequent knowledge mobility risk for the organisation.

Example 4. *Figure 7 considers a scenario whereby an organisation is analysing whether there are any knowledge risks related to dynamic web technologies (jsp and servlets). Using the ROC metric, Saviour and Shirley have been identified as potential risks in this regard. This is because both have strong knowledge of the two knowledge assets whilst Stephen and Sergio do not have any knowledge about them at all. One can also note that the fact that Stephen and Sergio do not know anything about the knowledge assets in question results in them not being marked as risks at all. Reason being that in this limited context, if either Sergio or Stephen leave, the organisation will not loose and valuable knowledge.*

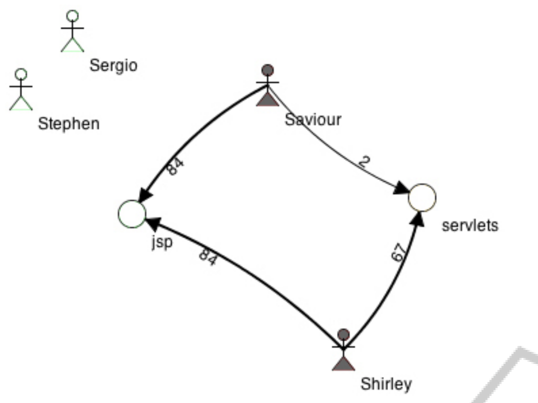

Figure 7: Analysis of mobility risk from a person perspective with regards to the knowledge assets jsp and servlets.

4.3 Knowledge Transfer Risk

The technique presented is also useful for identifying knowledge transfer risks. That is to say, situations which might compromise the likelihood of a knowledge transfer activity to be successful. Knowledge transfer activities are amongst the most important activities which enable organisations to hold on to organisational knowledge despite staff turnover. Although research carried out seems to indicate that most knowledge transfer success factors can only be detected and influenced by company culture and management practices, Cummings and Teng (Cummings, 2003) developed a research model consisting of nine key factors which affect knowledge transfer, two of which can be detected using the event-based cartographic approach proposed in this paper. These are *knowledge embeddedness* and *knowledge distance*. The former refers to the extent to which a particular knowledge asset is linked other knowledge assets within the organisational knowledge landscape. Cummings and Teng found that the more embedded a knowledge asset is, the more difficult it is to transfer. *Knowledge Distance* refers to the difference between two people's knowledge in relation to the knowledge asset being transferred. If two people share a relevant common basis of knowledge, their knowledge distance is said to be small. This makes a knowledge transfer exercise between such persons more likely to succeed than if they had a large knowledge distance separating them.

Both *knowledge embeddedness* and *knowledge distance* can be inferred from knowledge models constructed using the techniques presented in this paper.

4.3.1 Embeddedness

The following measure of embeddedness is propopsed:

$$embeddedness(\mathrm{k}) = |\mathrm{dep}(k)|$$

Where **dep**(k) is a function that given a knowledge asset k, returns the set of all knowledge assets which k *depends on*. Please note that since dependency is transitive (see section 4.1), this set will also include knowledge assets which are depended on by direct dependencies of k. Also, since composition is a form of dependency (see section 4.1), **dep**(k) will also return elements which k forms part of.

For any knowledge asset k, the higher the value of embeddedness(k), the more knowledge transfer risk k exhibits.

4.3.2 Knowledge Distance

Consider a knowledge transfer exercise whereby a person who is a knowledge source p_{src} is transferring a knowledge asset k to a second person who is a knowledge receiver p_{rec}. The knowledge distance measure takes into account p_{rec}'s lack of knowledge of all assets in **dep**(k) (see section 4.3.1):

$$\mathrm{distance}(p_{src}, p_{rec}, k) = \sum_{\forall k' \in dep(k)} \mathrm{dist}_{abs}(p_{src}, p_{rec}, k')$$

Where $\mathbf{dist}_{abs}(p_{src}, p_{rec}, k)$ returns the difference between the magnitudes of the knowledge relationships $p_{src} \xrightarrow{\text{knows}} k$ and $p_{rec} \xrightarrow{\text{knows}} k$. If the result is a negative number, **0** is returned. This is because we are only interested influencing knowledge distance in cases where the source person knows more than the receiver.

5 CONCLUSIONS AND FUTURE WORK

The work presented here has been trialled in an evaluation exercise whereby two development teams with four members each tracked their knowledge over a three month period. At regular intervals during this period, team members met with a researcher to analyse their knowledge model with respect to its perceived realism and also with respect to any apparent knowledge risks. In the case where risks were identified, mitigating actions were taken and the results are promising. Unfortunately, due to paper length restrictions, a detailed analysis of the results is not possible here.

With regards to future work, the authors are currently involved in a research project which aims to build and evaluate a software development process that makes knowledge management activities business as usual. This contrasts with the current state of affairs in software engineering whereby development processes' management of knowledge basically involves codification strategies in the form of project documentation. Other ongoing work includes long-term industrial evaluation of the proposed technique as well as exploration of extensions such as incorporating knowledge asset priority into risk calculations, automated team building and the development of a toolset enabling further experiments in the field.

REFERENCES

Alavi, M. and Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136.

Bjørnson, F. and Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, 50(11):1055–1068.

Botafogo, R. A., Rivlin, E., and Shneiderman, B. (1992). Structural analysis of hypertexts: identifying hierarchies and useful metrics. *ACM Trans. Inf. Syst.*, 10:142–180.

Cummings, J. (2003). Transferring r&d knowledge: the key factors affecting knowledge transfer success. *Journal of Engineering and Technology Management*, 20(1-2):39–68.

Dingsøyr, T., Djarraya, H. K., and Røyrvik, E. (2005). Practical knowledge management tool use in a software consulting company. *Commun. ACM*, 48:96–100.

Duffy, N. (1999). Benchmarking knowledge strategy. *Leveraging Knowledge for Business Performance 1999: Knowledge In Action*.

Earl, M. (2001). Knowledge management strategies: Toward a taxonomy. *Journal of Management Information Systems*, 18(1):215–233.

Linberg, K. (1999). Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2-3):177–192.

Micallef, M. (2011). A language for modelling software engineering knowledge. Technical Report CS2011-02, Department of Computer Science, University of Malta. Available from `http://www.um.edu.mt/ict/cs/research/technical_reports`.

Nolan Norton Institute, T. (1998). *"Putting the knowing organization to value" white paper*. Nolan Norton Institute.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1):14–37.

Pressman, R. (1998). Fear of trying: The plight of rookie project managers. *IEEE Softw.*, 15(1):50–51,54.

Ramal, M. F., Meneses, R. d. M., and Anquetil, N. (2002). A disturbing result on the knowledge used during software maintenance. In *Proceedings of the Ninth Working Conference on Reverse Engineering (WCRE'02)*, pages 277–, Washington, DC, USA. IEEE Computer Society.

Szulanski, G. (1996). Exploring internal stickiness: Impediments to the transfer of best practice within the firm. *Strategic Management Journal*, 17:27–43.

Tiwana, A. (2000). *The Knowledge Management Toolkit: Practical Techniques For Building A Knowledge Management System*. Prentice Hall.

Zack, M. (1998). What knowledge-problems can information technology help to solve. In *Proceedings of the Fourth Americas Conference on Information Systems*, pages 644–646.