Identifying Communities of Practice through Ontology Network Analysis

Harith Alani, Srinandan Dasmahapatra, Kieron O'Hara, and Nigel Shadbolt,
University of Southampton

This article describes Ontocopi, a tool for identifying communities of practice by analyzing ontologies of relevant working domains. Ontocopi spots patterns in ontological formal relations, traversing the ontology from instance to instance via selected relations.

Communities of practice—groups of individuals interested in a particular job, procedure, or work domain—informally swap insights on work-related tasks, often through quick chats by the water cooler. They act as corporate memories, transfer best practice, provide mechanisms for situated learning, and act as foci for innovation.[1,2]

Increasingly, organizations are harnessing communities of practice to carry out important knowledge management functions.[3] However, a significant first step is identifying the community, which often doesn't designate itself as such, and its members, who don't know they belong! So, this step involves determining which people in a community of practice have common interests in particular practices or functions and producing sets or clusters of related individuals. Community identification traditionally demands heavy resources and often includes extensive interviewing.

In this article, we describe Ontocopi (Ontology-Based Community of Practice Identifier), a tool to help identify communities. Ontocopi lets you infer the *informal* relations that define a community of practice from the presence of more *formal* relations. For instance, if A and B have no formal relation but they have both authored papers with C (formal relation), they might share interests (informal relation). Because Ontocopi works in this way, we cannot claim without qualification that it identifies communities of practice. Significant informal relations might have little or no connection to the formal ones. Here, we refer to the networks uncovered by Ontocopi as *COPs* and to informal social networks as *communities of practice*. We work under the assumption that COPs are sometimes decent proxies for communities of practice.

Ontocopi

We developed Ontocopi within the Advanced Knowledge Technologies (AKT) project[4] for *ontology-based network analysis* (ONA), which finds sets of instances associated with a selected instance in a knowledge base. (See the related sidebar.) By ontology, we mean combining a taxonomic structure of classes and relations with the knowledge base that results from instantiating the classes with domain objects. If you suppose that such an ontology represents a domain's objects and relations, you can analyze the connections between the objects. Ontocopi uses ontological relations to discover connections between objects that the ontology only implicitly represents. For example, the tool can discover that two people have similar patterns of interaction, work with similar people, go to the same conferences, and subscribe to the same journals.

Using an ontology to analyze such networks provides you with semantics for classes and relations. So, during analysis, you can select targeted relations for the community of practice and increase their weight in the algorithm and assign low or zero weights to unimportant relations. However, choosing the ontology is an important step because its content determines ONA's effectiveness. For example, the papers people publish are likely to be important for determining their interests, but if their publications

1094-7167/03/$17.00 © 2003 IEEE
Published by the IEEE Computer Society

Ontology-Based Network Analysis

ONA, a general graph-based algorithm, analyzes an ontology by viewing its instances as a set of nodes joined by the relationships in which they participate. We associate a real numbered weight *x*(*i*) for each node (instance) *i* in the graph, and for every edge, we associate a weight *R*(*i, j*), representing the weight transferred from node *j* to node *i* due to a particular (binary) relation *R* that the edge represents. You can assign weights *x*(*i*) to solve the self-consistent set of equations obtained by requiring that they be determined solely by weight exchanges via the relational links as indicated. Because we are interested only in the relative weights and not the absolute values, we need only solve for *x*(*i*) modulo an overall numerical factor—so we have to solve the equation

$$\text{constant} * x(i) = \text{Ó}_j\, R(i, j) * x(j).$$

This is an eigenvector problem, and if we further require the weights to be positive and if the values *R*(*i, j*) are all the same sign, the solution is the eigenvector corresponding to the largest eigenvalue of matrix ***R*** (called the Perron eigenvector). If the matrix is irreducible, you obtain the solution by a simple iterative procedure—$R^n\mathbf{y}$ is approximately proportional to the vector of weights, **x**, for a large-enough natural number *n* and for an arbitrary vector **y** (which we take to be **1**, the vector with all entries 1). The values *x*(*i*) will then measure node *I*'s relative importance in this network. A wide range of applications that order network data build on this method.[1]

To find the COP of an individual represented as node *p*, we find the Perron eigenvectors of two matrices—the matrix with elements *R*(*i, j*) and the matrix with elements *R*(*i, p*) and *R*(*p, j*) set to zero. (In practice, these elements are set to a very small number (1 – *z*), where *z* is very close to 1, to prevent the matrix from becoming reducible.) The change in these sets of values *x*(*i*) in the two cases is a measure of *p*'s influence on each of the other participants *i* in the network the ontology represents. Because this requires a computation over all instances in an ontology, you must explicitly check the transitive closure of the relations (matrix ***R***'s irreducibility).

Note that this ONA characterization doesn't distinguish between people and other types of nodes; it is a perfectly general method. So, although communities of practice are primarily structured around people, they also include objects,[2] and a COP can be based on any instance. Also, ONA's usefulness depends on the ontology containing sufficient relevant and explicit information to help define the desired implicit relations.

As a first approximation, we implemented the iterative calculation ($R^n.\mathbf{1}$) as a local spreading-activation algorithm with an asynchronous update rule (as described in the algorithm section). This approximation emphasizes the changes in the *x* values that are first order in *z*, with further heuristic adjustments (such as the locking mechanism) based on experiments. This lets us interpret the results on the basis of the semantics encoded in the ontology depending on various procedural modifications, as we see in the main article.

References

1. L. Page et al., *The PageRank Citation Ranking: Bringing Order to the Web*, tech. report SIDL-WP-1999-0120, Stanford Univ., Palo Alto, Calif., 1999.
2. E. Wenger, R.L. McDermott, and W. Snyder, *Cultivating Communities of Practice*, Harvard Business School Press, Cambridge, Mass., 2002.

do not appear in the ontology, it cannot supply information about them.

Ontocopi provides an algorithmic instantiation of the general ONA method. It plugs into the Protégé-2000 ontology and knowledge base editor.[5] Our experiments used an ontology developed within the AKT project to describe the immediate academic domain of Southampton University's Electronics and Computer Science Department, including people, papers, projects, and conferences.[2]

Figure 1 shows the Ontocopi user interface. Panel A displays the ontology classes that users can select. Panel B displays instances of the selected class that the user can choose. Ontocopi finds the COP of the chosen instance. When the user clicks the Get COP button (Panel C), a spreading activation search on the ontology moves from the selected instance to other instances connected to it by the selected relations, up to a maximum number of links. Weights of linked instances are calculated and a table (Panel D) displays the results. (The following sections describe the rest of the interface and our exact method for COP calculation.)

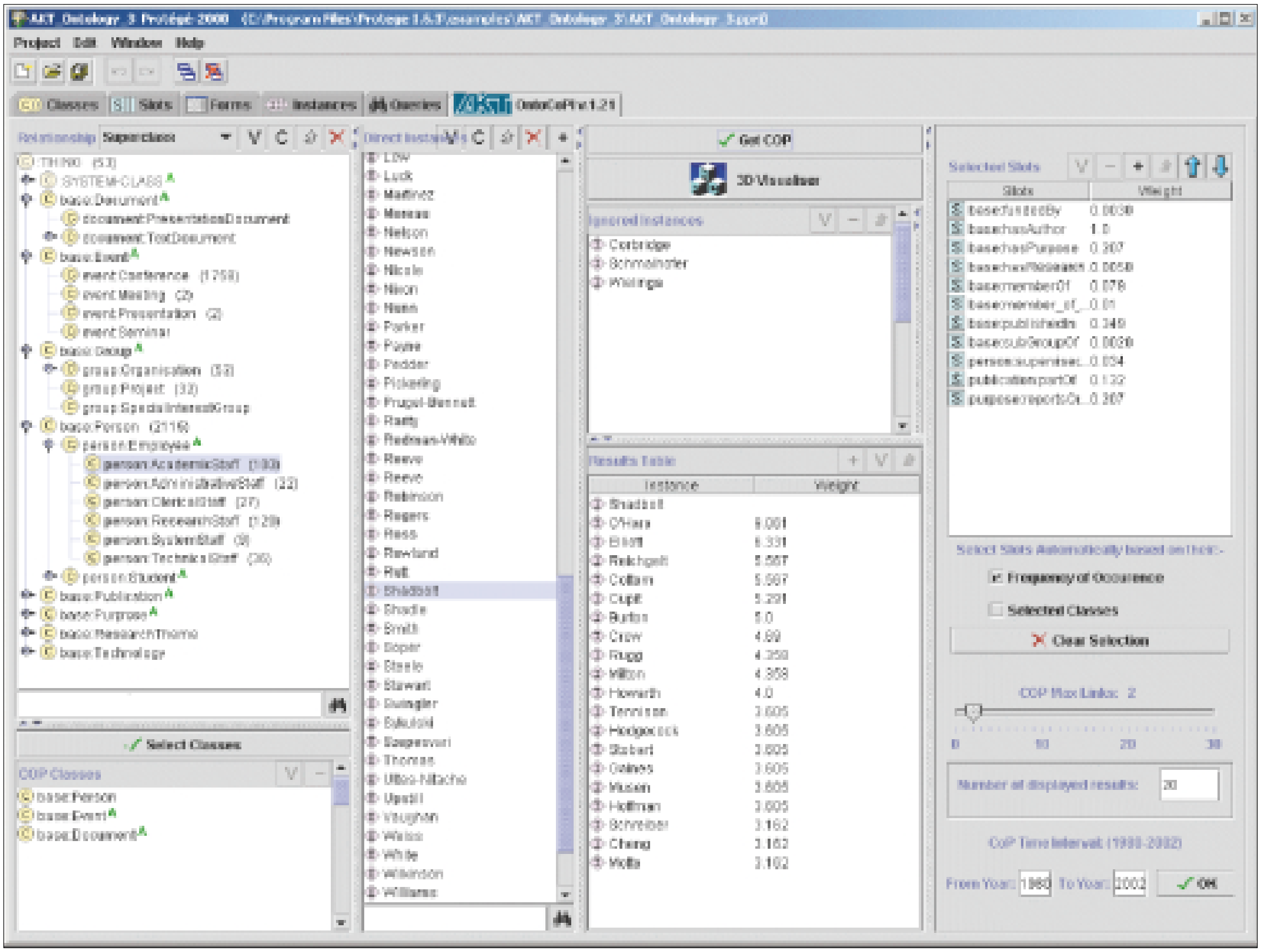
Selection modes

The system lets users select manual, automatic, and semiautomatic modes. For instance, in the display of relations and weights (Panel E), users can manually control weight allocation and alter them at any time. They can also select fully automatic or semiautomatic allocation.

Manual. The system lets users manually select relationships of interest and weight them by pressing the + button (Panel E). For example, if you are interested in people's collaborations on projects and coauthorships, you can select the relations *member of project, has author*, and *published in*. You can then set weights for these relations to increase or decrease their impact on the COP to be identified. The less weight you give a relation, the smaller its impact. This approach lets users completely control which relationships to traverse and how to weight them. But, you must know what the relationships represent and their relative importance. A relation's weight on the results not only proportionally affects the weight of other selected relations but also affects the number of these relations in the ontology. The more you use a relation, the greater its effect on results because it is traversed more often than other relations.

Automatic. The system can also select relations and calculate their weights automatically on the basis of how often the relations appear in the ontology, which indicates the relations' level of importance to that ontology and whether it provides good information about them. You can activate the fully automatic approach with the Frequency of Occurrence check box (Panel G).

Ontologies are usually populated unevenly. When an ontology is populated with instances, certain relations are normally used more than others, with some relations not used at all (say, the slot has been created but



Instance	Weight
Shadbolt	
O'Hara	6.081
Elliott	6.331
Reichgelt	5.567
Cottam	5.567
Cupit	5.291
Burton	5.0
Crow	4.69
Rugg	4.359
Milton	4.359
Howarth	4.0
Tennison	3.606
Hodgecock	3.606
Stobart	3.606
Oaines	3.606
Mussen	3.606
Hoffman	3.606
Schreiber	3.162
Cheng	3.162
Motta	3.162

Slots	Weight
base:fundedBy	0.0030
base:hasAuthor	1.0
base:hasPurpose	0.207
base:hasResearch	0.0050
base:memberOf	0.078
base:member_of_...	0.01
base:publishedIn	0.349
base:subGroupOf	0.0020
person:supervises...	0.034
publication:partOf	0.132
purpose:reportsOn...	0.207

Figure 1. The Ontocopi user interface. (See the main text for a description of the labeled panels.)

remains empty). This occurs because information is unavailable or because less effort was spent collecting it (a reflection of its relative importance). However, full automation can cause unevenness of representative data in the knowledge acquisition phase in some legacy knowledge sources.

So, fully automatic relation selection bypasses problems with user uncertainty, but frequency of use might only partially measure relevance to user interests.

Semiautomatic. The semiautomatic approach compromises by letting users select a concept of interest such as Person through the display (Panel F). This process limits automatic relation setting to only those used between specified concepts. You can activate it from the Selected Classes check box (Panel G). This approach lets users limit the COP to certain concepts when they're unaware of the underlying relationship modeling.

The algorithm

The Ontocopi expansion algorithm generates the selected instance's COP by identifying the set of close instances and ranking them by the weights they accumulate from several path traversals. It applies a breadth-first, spreading-activation search, traversing the semantic relations between instances (*ignoring directionality*) until it reaches a *link threshold*. This threshold is the maximum allowed number of consecutive links traversed between nodes, selected through Panel G.

The link threshold lets COPs have different ranges. A narrow COP might comprise only entities directly related to a project (project employees, member organizations, themes), while a wider COP might include indirectly related entities such as colleagues' coauthors or other projects about the same theme or subject. Figure 2 shows the pseudocode for Ontocopi's COP calculation algorithm.

Now consider the example network in Figure 3. Assume we need to identify query instance A's COP, using the relationships *hasAuthor, memberOf*, and *attended*, with the weights 1.0, 0.6, and 0.3. All instances have an initial weight of 1. Activation spreads from the query instance to neighboring instances in the network up to a given number of links. In the first expansion, query instance A passes on weights to all of its connected instances. The amount of weight passed equals the instance's weight multiplied by the traversed relationship's weight. In this case, A passes $1 * 0.6$ to D and $1 * 1$ to H. We add these to their initial weights of 1. In return, these instances pass their total weights to all their neighbors, so D, for example, passes $(1 + 1 * 0.6) * 0.6$ to B and A. Expansion stops when the link paths are exhausted or the link threshold is reached. (In the algorithm, locking and unlocking instances prevent feedback loops from continuing until it reaches the link threshold). We then raise results to the power $1/n$ to normalize them according to their link distance, where *n* is the minimum number of links traversed to reach the instance starting from the query instance. Instances therefore accumulate weight on the basis of the number of relevant relations they have with the initial instance.

The number of links to expand greatly affects the COP results because of the simplification introduced in this implementation. The algorithm attempts to identify the instances most like the query instance within a boundary the given link threshold defines. When we limit expansion to only one link, all identified instances have a direct relation to the query instance. As the number of links increases, so will the number of instances that have only an indirect link with the query instance.

In terms of its intellectual roots, the Ontocopi algorithm derives ideas from the literature on similarity measures and applies them to ONA. It builds on an approach introduced by Chris Paice,[6] in which relevance values of instances increase with the number of semantic paths leading to these instances. However, because you can represent ontological relationships bidirectionally (*has-author* versus *authored-by*), our algorithm differs from Paice's in that it ignores relationship direction. Furthermore, our algorithm lets an instance transfer some weight back to its "source" instance to ease a problem that arises when you apply Paice's method to a dense ontology: If activation spreads over more than a few links and reaches heavily connected instances (influential *hubs*), these instances receive disproportionately high weights accumulated from their numerous connections. Hence, our algorithm introduces a one-step-backwards weight transfer to give extra weight back to source instances. Nevertheless, you can propagate a high percentage of a hub's weight to some of its further connected instances, which in turn can earn an unjustified high COP ranking. To solve this, you can compensate by adjusting the weight passed from an instance based on the number of connections. The more connections an instance has, the more general it is and the less weight it can transfer.

Refining the picture

Getting the COP right depends on the ontology, the user's aims, and the domain. Even if rules of thumb emerged from long-term study of the technique, you would still need to experiment within any new domain to establish the ontology's network properties.

Let's assume the user wants to identify the COP of a person named Shadbolt, an instance of the class Academic Staff (as in Figure 1). The user can select relations and weight them manually or use the semiautomatic or automatic selection. Then, the user sets the link threshold and maximum number of COP results. The number of results implicitly sets a weight threshold to filter out any instance with a final weight less than the calculated value. This process also controls the number of results to display.

```
Initialise all instances weights to 1
Create a relationship-array of selected relationships and weights
Set query instance as the current instance
Mark current instance as unlocked and add it to an instance-array
Loop to the maximum number of links to traverse
    Search for the first unlocked instance in instance-array
    If found:
        Mark instance as locked
        Set instance as the current instance
        Get all instances connected to current instance with a relationship in the relationship-array
        Loop to number of connected instances
            If instance not in instance-array (new instance) and within given time-interval
                Weight of instance = initial weight + current instance
                                   weight * weight of connecting relationship
                Mark instance as unlocked and add it to instance-array
            If instance already in instance array
                Weight of instance = instance weight + current instance
                                   weight * weight of connecting relationship
        End loop
    If not found then exit
End loop
```

Figure 2. Pseudocode for the COP calculation algorithm.

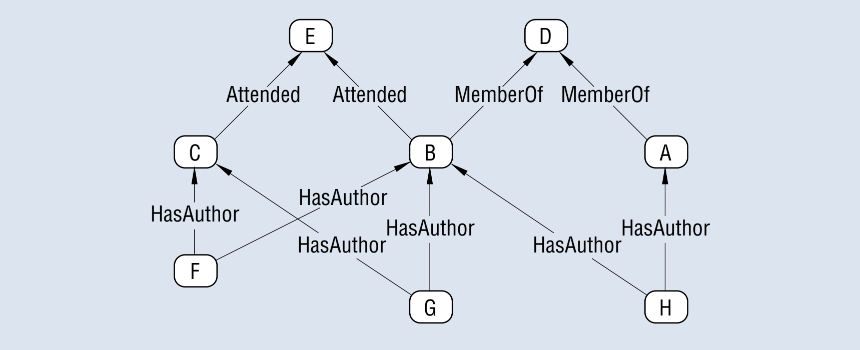


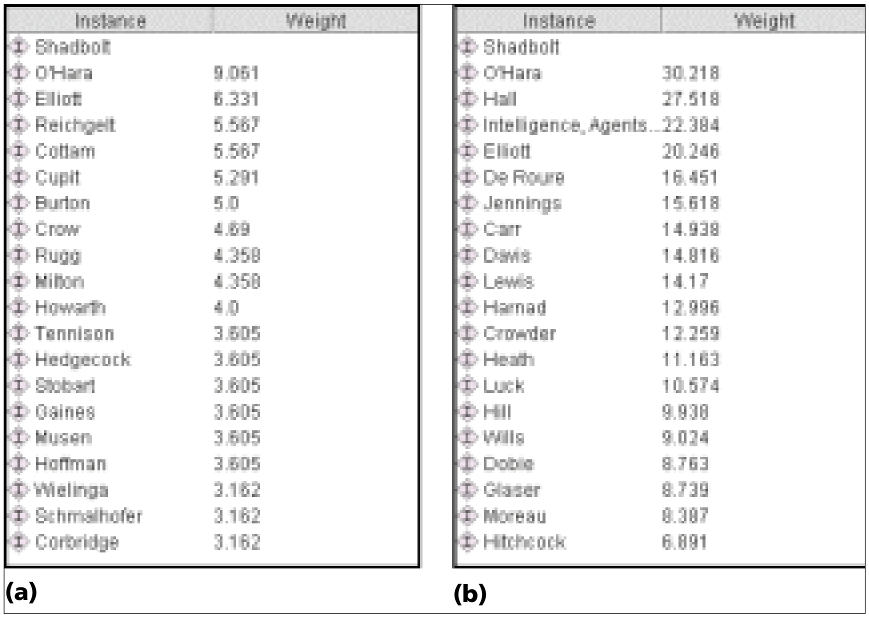
Figure 3. Example ontology network.

We describe the results of a set of experiments in which we used different settings to identify Shadbolt's COP in the AKT ontology. (We only considered the first 20 results of each experiment.)

Standard runs using automatic and manual settings

We first use a link threshold of 2 to identify Shadbolt's immediate COP. The automatic relations selector sets the highest weight of 1 to the relationship *hasAuthor*, which is why most objects in Shadbolt's COP are people and why the highly ranked people in Shadbolt's COP generally share the highest number of joint publications. Figure 4a shows that the closest person to Shadbolt is O'Hara, Shadbolt's trusty lieutenant, who works in the same department and has coauthored more than 30 papers with him.

Increasing the link threshold to 4, leaving the relation settings unchanged, gives the COP shown in Figure 4b. More instances are reached because of the analysis's extended range. Instances accumulate higher weights as more weights are passed around and new paths are explored. This wider COP includes instances indirectly connected to the query



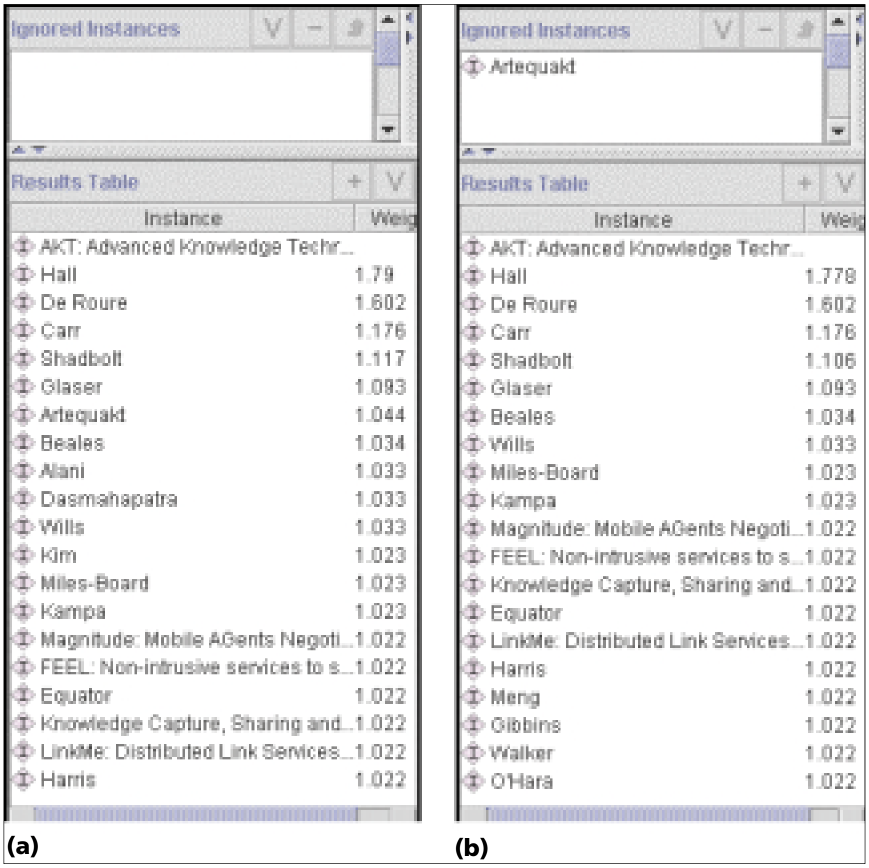
(a)

Instance	Weight
Shadbolt	
O'Hara	9.061
Elliott	6.331
Reichgelt	5.567
Cottam	5.567
Cupit	5.291
Burton	5.0
Crow	4.69
Rugg	4.358
Milton	4.358
Howarth	4.0
Tennison	3.605
Hedgecock	3.605
Stobart	3.605
Gaines	3.605
Musen	3.605
Hoffman	3.605
Wielinga	3.162
Schmalhofer	3.162
Corbridge	3.162

(b)

Instance	Weight
Shadbolt	
O'Hara	30.218
Hall	27.518
Intelligence, Agents...	22.384
Elliott	20.246
De Roure	16.451
Jennings	15.618
Carr	14.938
Davis	14.816
Lewis	14.17
Harnad	12.996
Crowder	12.259
Heath	11.163
Luck	10.574
Hill	9.938
Wills	9.024
Dobie	8.763
Glaser	8.739
Moreau	8.387
Hitchcock	6.891

Figure 4. Shadbolt's COP; automatic selection with (a) 2 links and (b) 4 links.



(a)

Instance	Weig
AKT: Advanced Knowledge Techr...	
Hall	1.79
De Roure	1.602
Carr	1.176
Shadbolt	1.117
Glaser	1.093
Artequakt	1.044
Beales	1.034
Alani	1.033
Dasmahapatra	1.033
Wills	1.033
Kim	1.023
Miles-Board	1.023
Kampa	1.023
Magnitude: Mobile AGents Negoti...	1.022
FEEL: Non-intrusive services to s...	1.022
Equator	1.022
Knowledge Capture, Sharing and...	1.022
LinkMe: Distributed Link Services...	1.022
Harris	1.022

(b)

Ignored Instances: Artequakt

Instance	Weig
AKT: Advanced Knowledge Techr...	
Hall	1.778
De Roure	1.602
Carr	1.176
Shadbolt	1.106
Glaser	1.093
Beales	1.034
Wills	1.033
Miles-Board	1.023
Kampa	1.023
Magnitude: Mobile AGents Negoti...	1.022
FEEL: Non-intrusive services to s...	1.022
Knowledge Capture, Sharing and...	1.022
Equator	1.022
LinkMe: Distributed Link Services...	1.022
Harris	1.022
Meng	1.022
Gibbins	1.022
Walker	1.022
O'Hara	1.022

Figure 5. COP based on Person and Project with (a) all instances considered and (b) the Artequakt project ignored.

instance through other instances—as in the case of supervisors of someone's coauthors. Hence, we see new people with fewer direct connections coming into the picture due to their connections with others; a COP with a higher link threshold can suggest COP memberships that an unaided subject would probably not ascertain.

To identify more specific COP types, the user can select the relations of interest and weight them manually or semiautomatically. For example, to identify Shadbolt's COP on the basis of his coauthors, project collaborators, and coworkers, the user can select the relationships *hasAuthor*, *memberOfProject*, and *memberOf* (that is, member of a department). The relationship weights must reflect the relative importance of the relationships to the COP the user seeks, and the user must allocate them manually.

Ignoring instances

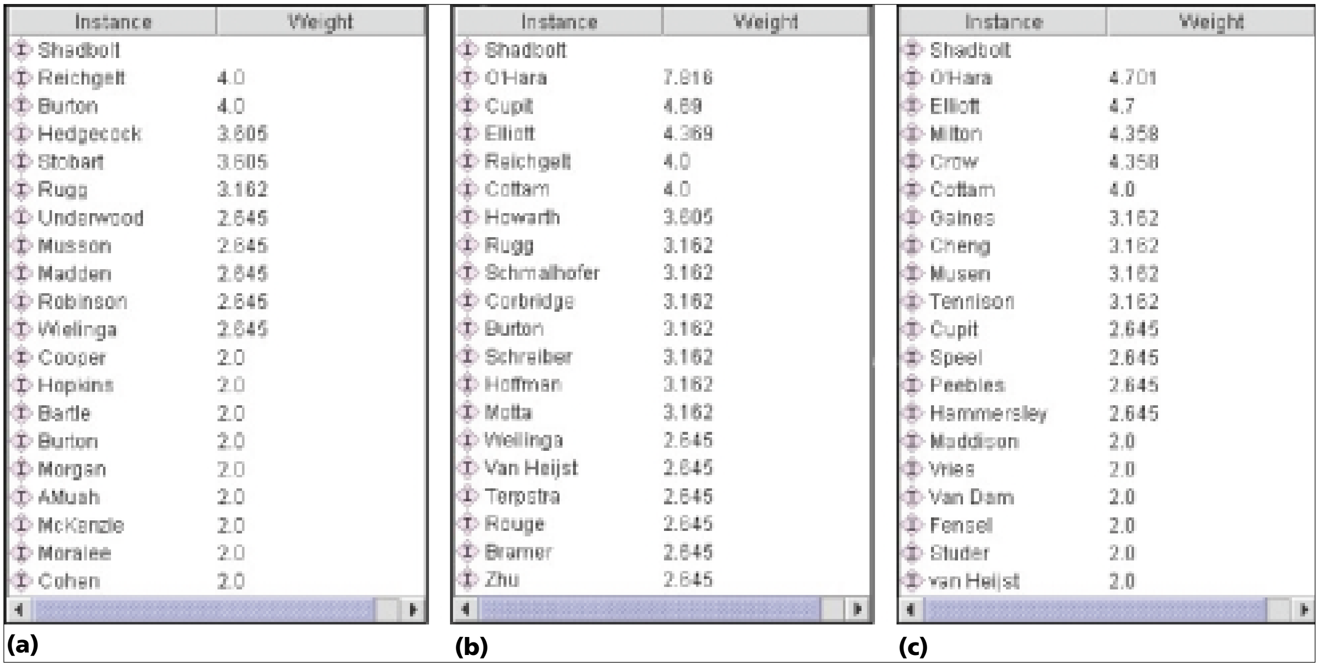
Studying the effect of losing an employee or project on the collaboration and communication lines between others in an organization helps uncover areas that such events could weaken. Identifying critical areas can help decision makers and planners maintain existing links and fill gaps.

To this end, Ontocopi lets users *ignore* specific instances when calculating a COP (Panel H in Figure 1). An ignored instance doesn't appear in a COP or on any of the relation paths, thus eliminating its effect entirely. Figure 5 shows the COP with respect to the concepts of People and Project; Figure 5a is the full COP, and Figure 5b is the COP calculated while ignoring a particular project, Artequakt. The COP differs slightly when Artequakt no longer exists. For example, Kim is not in the COP anymore. This indicates that the project will weaken or lose its connection with Kim (who is one of Artequakt's main contributors) if this project ends. The management team might prepare for this situation by creating new links to Kim before such an event occurs.

Temporally-based COP identification

The COPs Ontocopi uncovers are at best proxies for actual communities of practice. A clear difference between COPs and communities of practice is that the latter are dynamic—workers typically move in different communities as their working patterns and seniority levels change—and a COP is static (or at least as static as the ontology being used for analysis). For example, if a relationship changes—you stop working on a project—and the ontology is updated to reflect that, the COP will no longer have a record of the project.

However, if the ontology contains temporal information, Ontocopi can use it to present a more dynamic picture. For example, when the ontology represents the start and end dates of



(a)

Instance	Weight
Shadbolt	
Reichgelt	4.0
Burton	4.0
Hedgecock	3.605
Stobart	3.605
Rugg	3.162
Underwood	2.645
Musson	2.645
Madden	2.645
Robinson	2.645
Wielinga	2.645
Cooper	2.0
Hopkins	2.0
Bartle	2.0
Burton	2.0
Morgan	2.0
AMush	2.0
McKenzie	2.0
Moralee	2.0
Cohen	2.0

(b)

Instance	Weight
Shadbolt	
O'Hara	7.816
Cupit	4.69
Elliott	4.369
Reichgelt	4.0
Cottam	4.0
Howarth	3.605
Rugg	3.162
Schmalhofer	3.162
Corbridge	3.162
Burton	3.162
Schreiber	3.162
Hoffman	3.162
Motta	3.162
Wielinga	2.645
Van Heijst	2.645
Terpstra	2.645
Rouge	2.645
Bramer	2.645
Zhu	2.645

(c)

Instance	Weight
Shadbolt	
O'Hara	4.701
Elliott	4.7
Milton	4.358
Crow	4.358
Cottam	4.0
Gaines	3.162
Cheng	3.162
Musen	3.162
Tennison	3.162
Cupit	2.645
Speel	2.645
Peebles	2.645
Hammersley	2.645
Maddison	2.0
Vries	2.0
Van Dam	2.0
Fensel	2.0
Studer	2.0
van Heijst	2.0

Figure 6. Shadbolt's COPs: (a) 1985–1990, (b) 1991–1997, and (c) 1998–2002.

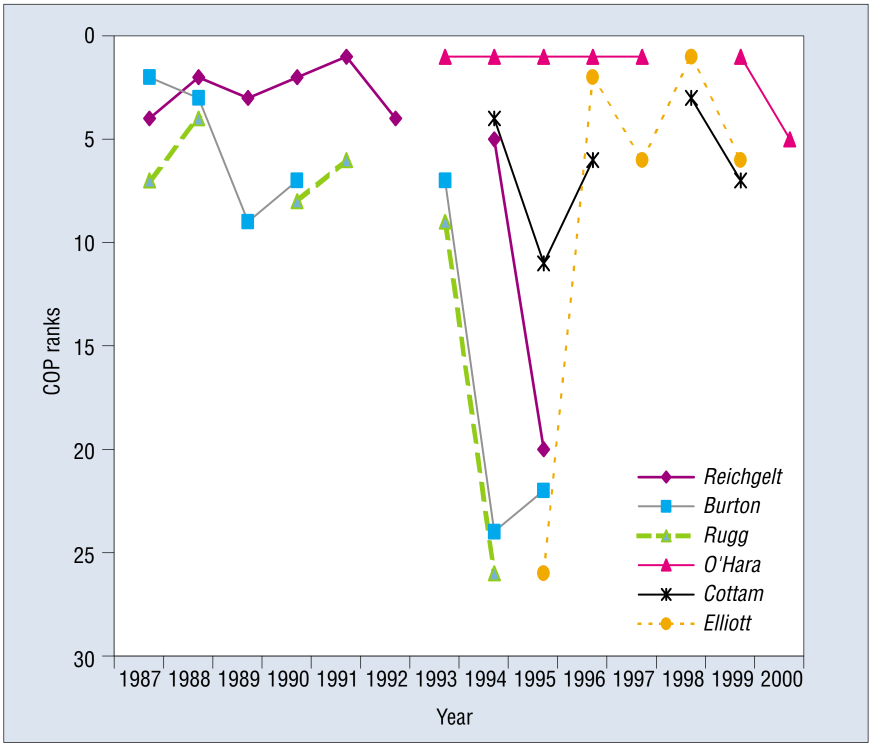


Figure 7. Changing ranks in Shadbolt's COPs.

a worker's employment on a project, you can set Ontocopi to focus only on relationships obtained within a specified pair of dates. Say you want to test the intuition that projects and coauthors in 2003 have better representation in the current community than those prior to 1990. When an ontology contains the information required to make the calculation, Ontocopi can produce a snapshot or a series of snapshots of a COP at a particular time.

Previous examples using the AKT ontology identified COPs using default temporal boundaries (from 1980 until 2002). But when we apply temporal limits (Panel G, bottom, in Figure 1), we can look at certain intervals' COPs. The COP algorithm checks whether each new instance is associated with a date. If that date falls outside the given temporal interval, it ignores that instance. Figure 6 shows Shadbolt's COPs in three different periods, focusing on coauthorship relations. Hedgecock, Stobart, and Underwood are highly ranked in Figure 6a but excluded from the COP in Figure 6b. Others such as Reichgelt, Burton, and Rugg are some of the most relevant to Shadbolt's COP in Figure 6a but fade gradually when their ranks drop in Figure 6b and disappear completely in Figure 6c. New people in the COP always replace the fading ones. For example O'Hara, Elliott, and Cottam appear in Figure 6b and maintain very high ranks in Figure 6c.

Figure 7 shows the time-related ranks of other people in Shadbolt's coauthorship-based COP. The rate of change depends, of course, on individuals fading from and moving into the COP. For example, Reichgelt climbed from fourth in 1987 to the top in 1991, then dropped until he disappeared for good in 1995. A new person, Elliott, joined this COP in 1995 and started to secure higher positions but also began to fade after 1998. Figure 7's discontinuous curves show that we had insufficient data for each person when some yearly calculations occurred.

Ontologies tend to lack temporal information because it's difficult to capture and represent such information. Some results of time-based COPs are inaccurate due to information loss. For example, even though the AKT ontology captures paper publication dates, they lack the start and submission dates. Nevertheless, when you track someone's career development, it can help to restrict ONA to a particular period. For example, to help find a new person for a certain job, you could calculate the predecessor's COP over the time in which he or she did that job.

Applications

COP detection can play an indirect role in numerous other management processes. The technique can extract implicit information from existing ontologies and play an interesting role in bootstrapping.

Organizational memory

A key problem in creating a usable organizational memory stems from its initial setup. When the OM contains little initial content, people can lack motivation to use it, producing a shortage of further content, cheap maintenance, and feedback. On the other hand, if the OM has a lot of content, users can overload on out-of-context information, and the large knowledge acquisition overhead generated by the content creation will increase the OM's initial costs.

However, when instantiated ontologies describe sufficient amounts of organizational structure, developers or users could use ONA (or Ontocopi) to create extra content.[7] An ONA algorithm could create interesting content on the fly, even during early stages of OM development. Users could easily employ ONA to suggest people to talk to about particular problems or issues that arise. If a user's question to the OM provided a starting node for an ONA, the analysis could provide related concepts and identify people strongly associated with the start node. In the context of information overload, users can always adjust ONA and alter their search criteria from the system defaults. Moreover, ONA remains context sensitive (where context is featured implicitly in the ontologies) but also generic. Furthermore, in ontology analysis, the user doesn't need to know anything about the ontology.

Recommender systems

A similar bootstrapping problem exists with recommender systems. These systems learn about user preferences for, say, Web pages over time and automatically find new pages similar to the user's historical preferences. In the Quickstep recommender system for online research papers, explicit feedback and browsed URLs help form user interest profiles, and the system computes a daily set of recommendations. The user can offer feedback to improve the training set and classification accuracy.

Our team has investigated integrating Quickstep and Ontocopi to help with the bootstrapping problem.[8] Upon startup, an ontology gives Quickstep an initial set of publications for each user. When a new user is added, the ontology provides this person's publication list (if those papers are already in the ontology), and Ontocopi provides a new COP by performing an ONA with the new user as a starting node. This COP (the most similar users) can then feed into a correlation between the new user's history of publication and similar user profiles to form the new initial profile.

Referential integrity

A third example application area for Ontocopi occurs within ontology development itself. As merging legacy ontologies, databases, and other information stores create more ontologies, problems will inevitably occur in preserving referential integrity across the merger. For example, you might refer to the same object or concept with different names (say, Alani, Harith, and H. Alani). We are investigating clustering potential instance duplicates in an ontology by using generic heuristics and soft string similarity measures and then using ONA to analyze the connections between the clustered instances.[9] Using them as the starting nodes of the analysis, we can calculate the instances' COPs within the ontology under construction itself and the degree of overlap between the COPs as a similarity measure. When the measure passes some threshold, it proves that the two instances, although represented by different names, are identical.

Thus far, we considered the results from analyzing transitive extensions of relationships to infer COP properties. But an ontology also provides metadata for interfacing with document repositories, which have their own sets of metadata, such as abstracts of publications. In our academic domain, we can use these sets to characterize COPs by finding word-use patterns in document abstracts authored by COP members. This information is not encoded in the ontology; we extract it by statistical analyses of the abstracts correlated with authorship details.

We start with a matrix of COP cooccurrences (by treating, say, the top five members of a COP as a single author) and words obtained from the abstracts. We then create a probabilistic model of this cooccurrence (using Hofmann's model[10]), which minimizes the statistical distance between the empirically observed data and a multinomial hidden variable model. The hidden variable identifies clusters of words related by common use. Briefly, the joint probability of an *event* of a word or author pair is given by

$$P(\text{word, author}) = S_z\, P(\text{word} \mid z)\, P(\text{author} \mid z)\, P(z),$$

where z indexes the hidden variable, and we derive the conditional probabilities on the right-hand side from the data.[10] The result is a set of probabilities of COP keywords that could help to independently characterize it. We might also use them to compare COPs or estimate the probabilities that someone would belong to one.

We would like to find better ways to present Ontocopi's results. At the moment, they appear as a list of instances, together with accumulated weights, which is only minimally enlightening. Researchers have experimented with more interesting visualizations,[2] which we could adapt. For example, when Ontocopi performs an analysis with an instance ignored, it could visually explain the conclusions. Or, to explain why instances accumulated their weights, 3D visualization might trace the shortest path from the initial instance to the target or represent the numbers in a calculation.

Because a COP is an artificial construct, its boundaries are not well defined. The user selects the number of instances to display, and the display cuts off at this point. We experimented with fixed-number and fixed-weight display models, and both have points for and against them.

With regard to identifying communities of practice, management itself lacks established methodologies, which obviously makes it difficult to integrate Ontocopi within a standard method. When we can precisely characterize Ontocopi's contribution to community management, it will be simpler to produce informative visualizations of results, and we are working on achieving this. Appropriate visualizations and the forms the explanations take depend on the application. We would like to research the sorts of queries that Ontocopi could answer and how to visualize them.

Finally, as we discussed, hubs bring in very large weights, distorting the COP. They also cause noise by connecting with irrelevant instances. We are trying to work out the best response. Should the activation stop spreading? We are identifying hubs and studying their effects in ONA's complete instantiation.

Although our weighting system lets us distinguish between relations of different significance, we have no easy way of doing this with instances. For example, it is probably more important from the COP point of view to be a member of a small group than of a large one (it says more about Alani that he is in the Intelligence, Agents, Multimedia Group, than that he is at the University of Southampton). We could use instance connectivity and class hierarchical level as indications of instance specificity.

Ontocopi is a flourishing prototype that taught us much about the properties of ontologies and communities of practice. We dealt with numerous issues and discovered many new interesting problems. Each new problem tells us more about communities and their representation; each new answer raises more problems. We hope that we have uncovered a virtuous circle that will help improve the management of communities within organizations. ■

The Authors

Harith Alani is a research fellow in the Department of Electronics and Computer Science, University of Southampton, UK. He is working on the Advanced Knowledge Technologies project, focusing on ontology development and applications and community of practice support tools. His other research areas include ontology network analysis, knowledge services, semantic similarity measures, and spatial information retrieval. He received a BSc in civil engineering from the University of Baghdad and an MSc and a PhD in computer studies from the University of Glamorgan in Pontypridd, Wales. Contact him at the Intelligence, Agents, Multimedia Research Group, Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; ha@ecs.soton.ac.uk; www.ecs.soton.ac.uk/~ha.

Srinandan Dasmahapatra is a lecturer in the Department of Electronics and Computer Science, University of Southampton. His research focus has shifted from exact solutions in statistical physics and quantum field theory in two dimensions to artificial intelligence, with an emphasis on knowledge representation and pattern recognition. He received a BSc from the University of Calcutta, India, and a PhD in physics from the State University of New York, Stony Brook. Contact him at the Intelligence, Agents, Multimedia Research Group, Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; sd@ecs.soton.ac.uk; www.ecs.soton.ac.uk/~sd.

Kieron O'Hara is a senior research fellow in the Intelligence, Agents, Multimedia Group, Department of Electronics and Computer Science, University of Southampton. His research interests include theoretical and applied epistemology; the sociology, politics, and economics of technology; and epistemological interpretations of conservatism in modern and classical thought. He has a DPhil in the philosophy of artificial intelligence and an MSc in computation from the University of Oxford, after studying logic and metaphysics at the University of St. Andrews. Contact him at the Intelligence, Agents, Multimedia Research Group, Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; kmo@ecs.soton.ac.uk.

Nigel Shadbolt is Professor of Artificial Intelligence in the Department of Electronics and Computer Science at the University of Southampton. His research interests range from biologically inspired robotics to knowledge-intensive systems research. He is the director of the Advanced Knowledge Technologies (AKT) project, a UK-funded project in which five universities are pursuing basic and applied research in knowledge management technologies. He studied philosophy and psychology at the University of Newcastle upon Tyne, UK, and obtained a PhD in artificial intelligence from the University of Edinburgh. He currently serves as editor in chief of *IEEE Intelligent Systems*. Contact him at the Intelligence, Agents, Multimedia Research Group, Dept. of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK; nrs@ecs.soton.ac.uk; www.ecs.soton.ac.uk/~nrs.

Acknowledgments

The Advanced Knowledge Technologies Interdisciplinary Research Collaboration, sponsored by the UK Engineering and Physical Sciences Research Council Grant GR/N15764/01, supports this work. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, and Southampton and the Open University. An earlier version of this article was presented at the 2002 IFIP Workshop on Intelligent Information Processing under the title "Ontocopi—Methods and Tools for Identifying Communities of Practice." We thank the workshop audience for their comments and Mark Musen for selecting this article for this special issue.

References

1. E. Wenger, R.L. McDermott, and W. Snyder, *Cultivating Communities of Practice*, Harvard Business School Press, Cambridge, Mass., 2002.

2. K. O'Hara, H. Alani, and N. Shadbolt, "Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition," *Proc. Conf. Int'l Federation Information Processing (IFIP)*, World Computer Congress, (WCC 2002), Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

3. R.G. Smith and A. Farquhar, "The Road Ahead for Knowledge Management: an AI Perspective," *AI Magazine*, Winter 2000, pp. 17–40.

4. AKT, "The AKT Manifesto," 2001; www.aktors.org/publications/Manifesto.doc.

5 M.A. Musen et al., "Component-based Support for Building Knowledge-Acquisition Systems," *Proc. Intelligent Information Processing* (IIP 2000) *Conf. Int'l Federation Information Processing (IFIP)*, World Computer Congress, (WCC 2000); http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-2000-0838.html.

6. C.D. Paice, "A Thesaural Model of Information Retrieval," *Information Processing and Management*, vol. 27, no. 5, 1991, pp. 433–447.

7. Y. Kalfoglou et al., "Initiating Organizational Memories using Ontology Network Analysis," *Proc. Knowledge Management and Organizational Memories Workshop*, 15th European Conf. Artificial Intelligence (ECAI), IOS Press, Amsterdam, 2002, pp. 79–89.

8. S. Middleton et al., "Exploiting Synergy Between Ontologies and Recommender Systems," *Proc. Semantic Web Workshop*, World Wide Web Conf., (WWW 02); http://eprints.ecs.soton.ac.uk/archive/00006487/03/www-paper.html.

9. H. Alani et al., "Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web," *Proc. 13th Int'l Conf. Knowledge Eng. and Knowledge Management* (EKAW 02), LNAI, vol. 2473, Springer-Verlag, Berlin, 2002, pp. 317–334.

10. T. Hofmann, "Probabilistic Latent Semantic Analysis," *Proc. 5th Conf. Uncertainty Artificial Intelligence* (UAI 99), Morgan Kaufmann, San Francisco, 1999, pp. 289–296.