ONTOCOPI

Methods and Tools for Identifying Communities of Practice

Harith Alani, Kieron O'Hara and Nigel Shadbolt
Intelligence, Agents, Multimedia Group
Dept. of Electronics and Computer Science
University of Southampton

Abstract: The paper describes ONTOCOPI, a tool for identifying communities of practice (COPs) by analysing ontologies of the relevant working domain. COP identification is currently a resource-heavy process largely based on interviews. ONTOCOPI attempts to uncover informal COP relations by spotting patterns in the formal relations represented in ontologies, traversing the ontology from instance to instance via selected relations. Experiments to determine particular COPs from an academic ontology are described, showing how the alteration of threshold and temporal settings, and the weights applied to the ontology's relations affect the composition of the identified COP.

Key words: ontologies, communities of practice, network analysis, distributed knowledge management, corporate memory

1. INTRODUCTION

Communities of practice (COPs) are informal self-organising groups of individuals interested in a particular practice. Membership is not often conscious; members will typically swap war stories, insights or advice on particular problems or tasks connected with the practice (Wenger 1998). An example of a COP might be the set of people in an organisation who do the same (or overlapping) jobs. They understand each other's problems, both with the job itself and with liaison with the outside world. A *de facto* community gradually emerges from their discussions and interests.

COPs can therefore take on a number of important roles for organisations. They may (a) act as corporate memories, (b) transfer best practice, (c) provide mechanisms for situated learning of the practice, and

(d) act as foci for innovation. For individuals, the COP promotes the smooth integration of the practice with daily working life. For an example where COPs have been exploited in knowledge management, see the experience of Schlumberger (Smith and Farquhar 2000).

However, COPs are difficult to identify within organisations – an essential first step to understanding the knowledge resources of an organisation (Wenger 1999, McDermott 1999). In this paper we describe ONTOCOPI, the ONTOlogy-based Community Of Practice Identifier, a tool which uses ontology-based network analysis to support the task of COP identification. As a first cut proxy for a COP, we look for the set of most similar instances to a selected instance in the knowledge-base (i.e the instances that have most in common). ONTOCOPI was built as part of the Advanced Knowledge Technologies (AKT 2001) project.

Considerations of space in this paper preclude a deep discussion of the theory underlying ONTOCOPI. For such a discussion see (O'Hara et al 2002).

The structure of the paper is as follows. Section 2 will briefly examine the issues relating to the use of ontologies of working domains to identify COPs, while section 3 will then set out the principles underlying ONTOCOPI. Sections 4 and 5 will then discuss the current and future refinements of ONTOCOPI's performance.

2. EXPLOITING ONTOLOGIES FOR COMMUNITIES OF PRACTICE

ONTOCOPI is a tool for *ontology-based network analysis* (ONA). By an ontology we refer to the classification structure and the knowledge-base of instantiations. If an ontology represents the objects and relations in a domain of work, then it can be analysed to extract the connections between entities in that domain. A COP is defined by certain relations between entities relating to that practice, and so the aim of ONTOCOPI is to extract patterns of such relations.

The advantage of using an ontology to analyse such networks is that relations have semantics or types. Hence certain relations – the ones relevant to the COP – can be favoured in the process of analysis. During the analysis the weight of the contribution made by the important relations is high, while that of the less important ones can be made relatively low, or zero.

We discuss ONA in more detail in (O'Hara et al 2002). There are some important points to note here, though. First, the effectiveness of ONA for COP identification is dependent to a large extent on the content on the ontology and the properties of the COP. The choice of ontology therefore is an important step.

The essence of a COP is that it is an informal set of relations; ontologies will be wholly or largely made up of formal relations. By 'formal' here we mean relations that are determinate, fixed and cheap to establish/monitor, such as the relation of being a member of a group, being the author of a paper, having a particular telephone number. By 'informal' we mean relations that are often indeterminate and expensive to establish, such as a tendency to have a drink together after work. The ONTOCOPI hypothesis is that such informal relations can be inferred from the presence of formal relations. For instance, if A and B have no formal relation, but they have both authored papers (formal relation) with C, then that indicates that they might share interests (informal relation).

3. ONTOCOPI

The AKT ontology is implemented in Protégé 2000 (Eriksson et al 1999). ONTOCOPI plugs into Protégé and uses the ontology as its raw material. For more on the AKT ontology, see (O'Hara et al 2002). The user interface is shown in *Figure 1*. The left hand panel shows the ontology, to allow the user to select a class; second on the left displays the class instances, allowing an instance to be selected. Top right shows the available relations. The user selects the relations that he feels will be important in COP identification, and gives these weights depending on their relative importance (this can be done automatically – see below); selected relations and weights are displayed in the middle right panel. Controls at the bottom right allow the user to determine threshold and temporal settings.

When the user clicks the 'Get COP' button, 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 set as part of the threshold settings (see Section 3.2). Weights of linked instances are calculated and results are displayed in the third column. Currently there is no restriction of the type of object that can appear in a COP. One may want to find the COP of a person, and the COP may largely be made up of instances of the class 'person'. However, it may be desired to find the COP associated with, say, a particular journal, or subject area, or research group; type restrictions would prevent a search being done on such items.

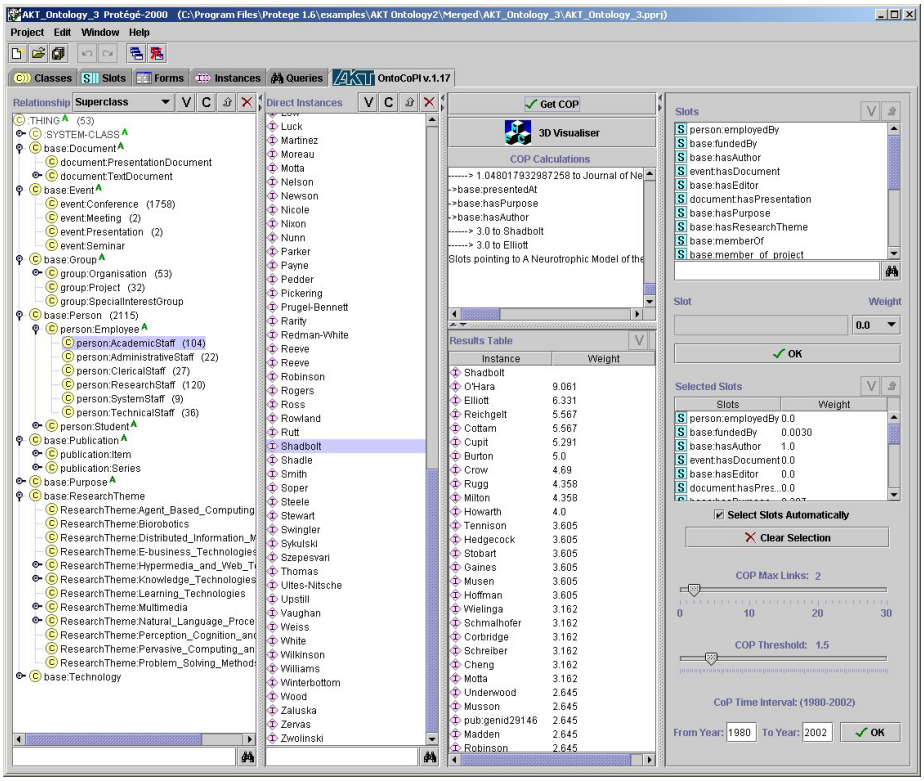


Figure 1. ONTOCOPI's User Interface

3.1 Relation and Weight Selection

Selecting the relations and weights can be manual or automatic. The system allows the user to select the relationships of interest, and weight them as needed. For example if the user is interested in peoples' collaboration on projects and co-authorships, then the relations *memberOfProject*, *hasAuthor*, and *publishedIn* can be selected. The user can then set weights for these relations to increase or decrease their impact on the COP to be identified. The less weight a relation is given, the less its impact will be.

The advantage of this approach is that users have total control on which relationships to traverse and how they should be weighted. But the user needs to know what the relationships represent, and have an idea of how important they are for his purposes. The effect of a relation's weight on the results is not only proportional to the weight of the other selected relations, but also on the number of these relations in the ontology. In other words, the more a relation is used, the greater its effect on results because it will be traversed more often than other relations.

The system can also select relations and calculate their weights automatically based on the frequency of use of these relations in the ontology, which is taken as an indication of the level of importance of those relations to that ontology, and whether the ontology is good or weak in providing information related to certain relationships. Ontologies are normally unevenly populated. Our experience shows that when an ontology is populated with instances, certain relations will normally be used more than others; some relations might not be used at all (i.e. the slot has been created but not filled in for any instance). This is normally due to the unavailability of certain information, or that different information has different levels of importance reflected in the amount of effort given to collecting and adding it to the ontology. This approach of selecting relations bypasses problems about user uncertainty, but equal frequency of use may only be a partial measure of its relevance to a user's interests.

3.2 The Algorithm

The expansion algorithm generates the COP of the selected instance (this is a person in our experiments, but could be any type of object) by identifying the set of close instances and ranking them according to the weights of their relations. It applies a breadth first, spreading activation search, traversing the semantic relations between instances (ignoring directionality) until the link threshold is reached. Starting with a weight of 1 for all instances, it transfers weights to all other instances following a set of weighted relations. The pseudocode is given in *Figure 2*; where n is the number of links traversed to reach the instance starting from the primary instance.

Consider the example in *Figure 3*. Assume we need to identify the COP of the query instance *A*, using the relationships *hasAuthor*, *memberOf* and *attended*, with the weights 1.0, 0.6, and 0.3 respectively. All instances will have an initial weight of 1. Activation will spread from the query instance to neighbouring instances in the network, up to a given number of links. In the first expansion, the query instance *A* will pass on weight to all the instances it is connected to. The amount of weight passed equals the weight of the instance multiplied by the weight of the traversed relationship. In this case, *A* passes 1*0.6 to *D*, and 1*1 to *H*. These will be added to their initial weights of 1. In return, these instances will pass their total weights to all their neighbours, so *D* for example will pass (1+1*0.6)*0.6 to *B* and *A*. Expansion will stop when the link paths are exhausted or the link threshold is reached (in the algorithm, locking/unlocking instances prevent feedback loops continuing till the link threshold is reached). Results are then raised to the power 1/n to normalise 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 based on the number of relevant relations they have with the initial instance.

```
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)
         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
Normalise all weights to the power 1/n
Rank instance-array according to instance final weights
```

Figure 2. The ONTOCOPI Algorithm in Pseudocode

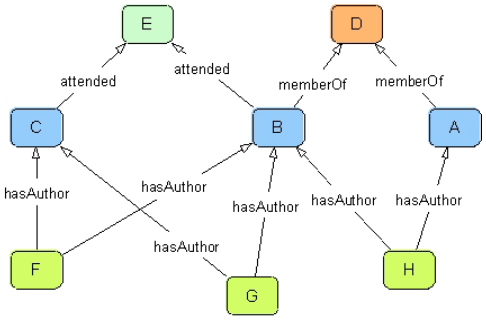


Figure 3. Example Ontology Network

The number of links to expand has an important effect on the COP results. The algorithm attempts to identify the instances with most in common with the query instance within a boundary defined by the given link-threshold. If expansion is limited to one link only then all identified instances will have a direct relation to the query instance. As the number of links increases, so will the number of instances which only have an indirect link with the query instance.

The number of indirectly-linked instances will also increase if *hubs* appear in the connected network. Hubs are highly-connected instances, and are very influential. They often score high ranks in the identified COPs as they receive high weights accumulated from their many connections. The drawback of this is that a high percentage of a hub's weight can be propagated to some of its connected instances which in turn will earn an 'unjustified' high rank in the COP. One possible approach to take is to compensate the weight to be passed from an instance based on the number of connections the instance has. The more connections an instance has, the more general it is considered to be, and hence the less weight it can transfer.

In terms of its intellectual roots, the ONTOCOPI algorithm takes ideas derived from the literature on similarity measures and applies them to the context of ONA. It builds on an approach introduced by Paice (1991) where relevance values of instances increase with the number of semantic paths leading to these instances. The algorithm however is different to Paice's in that relationship direction is ignored, since ontological relationships can be represented bi-directionally (e.g. *has-author* vs *authored-by*).

Furthermore, ONTOCOPI's algorithm allows an instance to transfer some weight back to its "source" instance, to ease a problem that arises when applying Paice's method to a dense ontology, were some instances have large number of connections. If activation is spread over more than few links, reaching heavily connected instances, then such instances will receive disproportionately high weights accumulated from their large number of connections. Hence a one-step backwards weight transfer is introduced in our algorithm to give extra weight back to source instances. We are experimenting with other alternatives, such as applying a weight-transfer cost for each instance based on its number of connections.

4. REFINING THE PICTURE

Getting the COP right will depend on the ontology, on the purposes of the user, and on the domain. Even if there are rules of thumb that emerge from study, experiments would still need to be carried out in any new domain to establish the network properties of the ontology. In this section, we discuss some of the experiments tried on the AKT ontology.

Lets assume the user would like to identify the COP of *Shadbolt*, an instance of the class *Academic Staff*. He can select relations and weight them manually, or go for the automatic selection. The selected relations and their weights will be displayed in the *Selected Slots* table on the right panel. Link and weight thresholds are set with slide bars. The weight threshold is used to filter out any instance with a final weight less than the given value. This is used to reduce the amount of noise in the results set, which often occurs

when expanding to a high number of links, or if the initial instance has low connectivity (i.e. not much information is available about the selected instance). The weight threshold also allows the user to control the display of results to, for example, only highly ranked entities if the interest is to identify strongly related entities only. We can describe the results of a set of experiments with ONTOCOPI using different settings to identify the COP of *Shadbolt* based on the AKT ontology. Note that only the first 20 results of each experiment are displayed.

4.1 Using Automatic Settings

The extent of the identified COP is dependent on the number of links to expand from the COP query instance. Here we first use a 2 link-threshold to identify the immediate COP of *Shadbolt*; the automatic relations selector was used, which sets the highest weight of 1 to the relationship *hasAuthor*, which is the reason why the highly ranked people in *Shadbolt's* COP are in general the ones with the highest number of joint publications with him. It can be seen from *Figure 4*(a) that the closest person to *Shadbolt* was found to be *O'Hara*, *Shadbolt's* trusty lieutenant, who works in the same department, and has co-authored more than 30 papers with him.

Instance	Weight
Shadbolt	
O'Hara	30.218
Hall	27.517
Intelligence, Agent:..	22.36
Elliott	20.246
De Roure	16.45
Jennings	15.617
Carr	14.937
Davis	14.816
Lewis	14.169
Hamad	12.996
Crowder	12.258
Heath	11.162
Luck	10.574
Hill	9.938
Wills	9.023
Dobie	8.762
Glaser	8.739
Moreau	8.387
Hitchcock	6.89

(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)

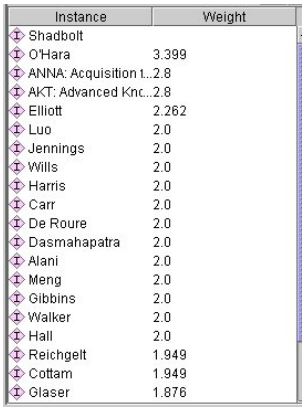
Figure 4. Shadbolt's COP, Automatic Selection (a) 2 links (b) 4 links

Increasing the link threshold to 4, with the relation settings unchanged, gives the COP shown in *Figure 4*(b). More instances have now been reached as the range of analysis is extended. Instances have now accumulated higher weights as more weights are passed around and new paths are explored. This COP is wider than before and it includes instances that are indirectly connected to the query instance through other instances, for example the

supervisors of someone's co-authors. Hence we see new people with less direct connections coming into the picture because of their connections with others; in this way we can see that the COP identified with a higher link threshold can make suggestions for COP membership that the unaided subject would be less likely to come up with.

4.2 Using Manual Settings

To identify more specific types of COP, the user can select the relations of interest and weigh them manually. For example to identify the COP of *Shadbolt* based on his co-authors, project collaborators, and co-workers, then the relationships *hasAuthor*, *memberOfProject*, and *memberOf* can be selected. Using the relationship weights of 0.2, 0.9 and 0.3 respectively, the resulting COP will be as in *Figure 5*.



Instance	Weight
Shadbolt	
O'Hara	3.399
ANNA: Acquisition t...	2.8
AKT: Advanced Knc...	2.8
Elliott	2.262
Luo	2.0
Jennings	2.0
Wills	2.0
Harris	2.0
Carr	2.0
De Roure	2.0
Dasmahapatra	2.0
Alani	2.0
Meng	2.0
Gibbins	2.0
Walker	2.0
Hall	2.0
Reichgelt	1.949
Cottam	1.949
Glaser	1.876

Figure 5. Shadbolt's COP, manual selection

This COP differs from the one identified in the previous section as some of the instances in this COP (all the ones with a weight of 2) have no joint papers with *Shadbolt*, but are all members of the same project, group, and department. The results will obviously be slightly different if the weights of the selected relationships change. For example people with more joint publications with *Shadbolt* will get higher values if the weight of *hasAuthor* is increased.

Specifying certain relationships to be used by ONTOCOPI needs some understanding of their semantics. It is our intention to facilitate this task by allowing the user to select the main concepts of interest from which the system can select and weight relevant relationships automatically.

4.3 Temporally-based COP Identification

Previous examples identified COPs using default temporal boundaries (from 1980 till 2002). Temporal limits can be applied to restrict COPs to certain intervals. Figure *6* shows the COPs of *Shadbolt* in three different periods, focusing on co-authorship relations. *Hedgecock*, *Underwood*, and *Stobart* were highly ranked in (a) but were excluded from the COP in (b). Others such as *Reichgelt*, *Burton*, *Rugg* were some of the most relevant to Shadbolt's COP in (a) but faded gradually when their ranks dropped in (b) and disappeared completely in (c). There are always new people in the COP replacing the fading ones, for example *O'Hara*, *Cottam*, and *Elliott* appeared in (b) and maintained very high ranks throughout (b) and (c).

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
AMuah	2.0
McKenzie	2.0
Moralee	2.0
Cohen	2.0

(a)

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
Weilinga	2.645
Van Heijst	2.645
Terpstra	2.645
Rouge	2.645
Bramer	2.645
Zhu	2.645

(b)

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

(c)

Figure *6*. Shadbolt's COPs, (a) 1985-90, (b) 1991-7, (c) 1998-2002

The Proustian *Figure 7* shows the time-related ranks of certain people in *Shadbolt's* co-authorship-based COP, displaying how people fade out of the COP while others move in. The rate of change in a COP depends of course on the movements of these individuals. For example *Reichgelt* climbed from 4th in 1987 to top in 1991, then dropped until he disappeared for good in 1995. A new person, *Elliott*, joined this COP in 1995 and started to climb and secure higher positions but also began to fade after 1998.

Time-based COP identification can be improved if more temporal information is available. Ontologies tend to lack temporal information due to the difficulty in capturing such information and the complexity of representing it. Some of the results of time-based COPs cannot be very accurate due to information loss. For example even though the date when papers were published is captured in the AKT ontology, it is not known when the work on these papers actually began, or when they were submitted.

As COPs are highly variable through time, it would be useful to be able to filter out relations that did not obtain in particular relevant periods; however, such information must be present in the ontology in the first place.

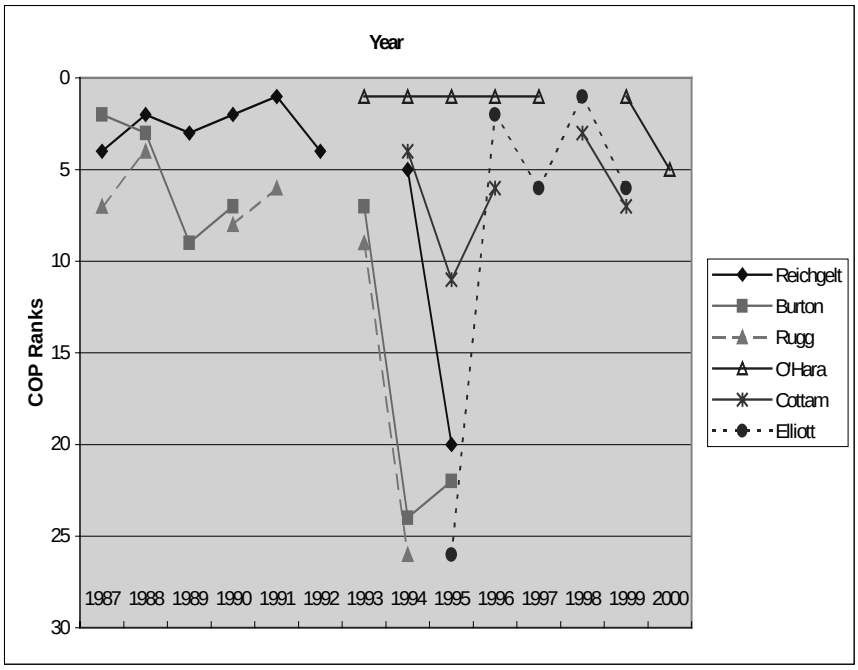


Figure 7. Changing Ranks in Shadbolt's COPs

5. DISCUSSION

By using ontologies, ONTOCOPI makes itself sensitive to the semantics of the relations out of which networks are built. As a support tool for COP identification, it has the scope to cut down search spaces radically.

This contrasts with similar work investigating networks of semi-formal relations between people or within systems. For instance, the analysis of networks of pages and hyperlinks to identify hubs and authoritative sites on the web in order to improve search engine results (e.g. Page et al 1999) is based on the number and direction of links between web pages, but the significance of such links is lost after a page is authored. Work related to ONTOCOPI is discussed in more depth in (O'Hara et al 2002).

We have discussed ONTOCOPI's ontology-based analysis techniques; we should also express the following caveats (cf. O'Hara et al 2002).

1. ONTOCOPI makes an explicit assumption that (some) informal COP relations can be inferred from the formal ones in an ontology.

2. In any new domain, a range of trials would have to be carried out to determine the interesting link thresholds and relation weights.
3. Note the problem of *brokers* or *boundary objects* (people or objects who exist in two COPs). In cases such as these the COPs identified may be the *union* of two or more COPs. It could be that this is a widespread problem, though Wenger (1998) does not think so. Much will depend on what possible filtering information is represented in the ontology.

Future research will focus on ways of filtering out noise and making search more flexible. Further scenarios will also be employed to ascertain which other knowledge management tasks can exploit ONA (e.g. coreference identification).

ACKNOWLEDGEMENTS

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

REFERENCES

AKT, The AKT Manifesto, 2001, http://www.aktors.org/publications/Manifesto.doc

Eriksson H, Fergerson R, Shahr Y, Musen M. Automatic generation of ontology editors. Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modelling and Management (KAW'99); 1999; Banff. Canada.

McDermott R., Why information technology inspired but cannot deliver knowledge management, California Management Review, 1999; 41

O'Hara K, Alani H, Shadbolt N. Identifying communities of practice: analysing ontologies as networks to support community recognition. Proceedings of the World Computer Congress; 2002; Montreal. Canada.

Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing Order to the Web. Stanford University working paper SIDL-WP-1999-0120; 1999.

Paice C.D. A thesaural model of information retrieval. Information Processing and Management 1991, 27(5):433-447

Smith R.G., Farquhar A. The road ahead for knowledge management: an AI perspective. AI Magazine, Winter 2000, 17-40

Wenger E, *Communities of Practice: Learning, Meaning and Identity*. Cambridge University Press, Cambridge, 1998.

Wenger E., Communities of practice: the key to knowledge strategy. Knowledge Directions: The Journal of the Institute for Knowledge Management, 1999, 1:48-93