Past and future of reflective middleware: Towards a corpus-based impact analysis

François Taïani, Paul Grace, Geoff Coulson, Gordon Blair
Lancaster University, Lancaster LA1 4YW, UK
{f.taiani,p.grace,g.coulson}@lancaster.ac.uk, gordon@comp.lancs.ac.uk

ABSTRACT

Analysing the impact of novel middleware abstraction is a crucial activity, in particular when applied to reflective middleware, a paradigm that was proposed 10 years ago. Impact analysis is unfortunately a difficult and multifaceted task. We argue here for a broad and systematic approach based on statistical text analysis. We present preliminary results that demonstrate the potential of this approach to uncover hitherto unsuspected trends, and hopefully inform the future of reflective middleware research.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]; I.2.7 [**Natural Language Processing**]: [Text analysis]

Keywords

middleware, reflection, survey, impact, corpus linguistics

1. INTRODUCTION

The application of reflection to middleware was proposed 10 years ago [2, 12, 13] to ease the development, configuration, adaptation, and evolution of distributed systems.

This approach has since then given rise to a vibrant corpus of research, and a number of surveys have reviewed the resulting research, either as their main topic [1, 11], or in the context of a particular concern or area [4, 6]. To the best of our knowledge, however, none of them provided a comprehensive assessment of the impact reflection has had on middleware research. Assessing the impact of a seminal research idea is generally useful. It is even more so in the context of middleware research, as middleware aims for a large part at proposing abstractions to facilitate the development of complex distributed systems. The quality of an abstraction is difficult to evaluate precisely, but we would argue that the response of other researchers is symptomatic of an abstraction's significance and ramifications.

Assessing the research impact of reflective middleware is unfortunately a daunting task: What criteria to use? What data to consider? How should evidence be drawn? How to avoid data explosion? In this article we report on our early attempts to address some of these issues using techniques derived from corpus linguistics. In doing so, we have tried to chart the landscape of reflective middleware research in a slightly unconvential way, with the goal of uncovering unsuspected trends, gaps, and connections. With this work we hope to contribute to the community's ongoing reflection on what has been achieved so far, and (hopefully) provide new ideas for future research.

The article is organised as follows: Section 2 presents our approach along with some background on corpus comparison. Section 3 describes our results. Section 4 discusses the benefits and limitations of our experiment. Finally, Section 5 outlines related work, and Section 6 concludes.

2. APPROACH AND BACKGROUND

Our first step was to select a representative set of articles related to research on reflective middleware. In this preliminary work we focused on the 1998 article by Blair et al. *An architecture for next generation middleware* [2] as the starting point of our study. With 310 citations this article is according to *Google Scholar* [9] the second most cited paper of the middleware conference series. From this set of 310 papers, we selected the 40 most quoted articles that are conference or journal articles written in English. This led us to discard one article in Chinese (in 19th position with 53 citations) and one Ph.D. thesis (in 39th position, with 29 citations).

This mode of selection is of course quite questionable, as many works related to reflective middleware do not cite Blair's 1998 paper! (Some of our own indeed fall in this category.) Limiting oneself to one paper might further bias the sample in one particular research direction, or close-knitted community. These are all valid criticisms, and we agree that any full-fledged study should rely on a more comprehensive selection. (We return to these points in Section 4.) However, we also feel this sample is representative *enough* to highlight the possibilities of keyword-analysis, as long as one keeps in mind its limitations.

As can be seen on Figure 1 (*all articles*), the 310 articles that cite [2] span 10 years of research, and were themselves highly quoted (Figure 2)[1]. Figure 1 shows that although the sample of 40 papers we selected only covers the years 1998-2004 (*top 40*), they roughly mirror the general publication trend of these years (*all articles*). According to Figure 2, these 40 top

[1] 22 articles had no publication year and are not shown on the figures.

ARM 2008, December 1, 2008, Leuven, Belgium.

papers generated most of the collective impact (measured in terms of citations) during 1998-2004, with their contribution hovering between 58% (2004) and 98% (1998).

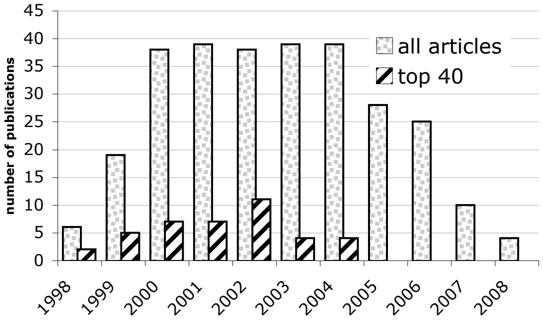

Figure 1: Yearly distribution of articles citing [2]

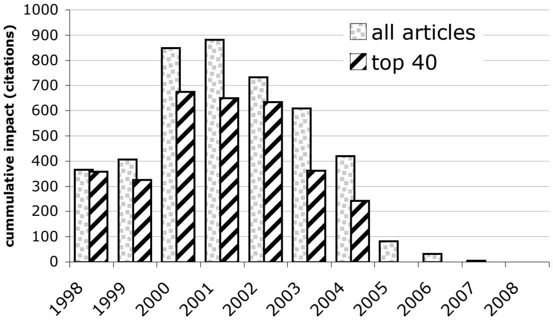

Figure 2: Cumulative citation count of articles citing [2] by year of publication

Using these 40 papers as a representative corpus of reflective middleware research over the years 1998-2004, we followed a three steps approach to analyse their content:

- **Keyword extraction:** We used a technique known as *corpus comparison* [16] to extract relevant keywords.
- **Correlation analysis:** We then computed correlation coefficients between each pair of keywords to detect linear correlations.
- **Trend analysis:** Finally we computed correlation coefficients between the density of each keyword and the year of publication to detect trends in the emergence or disappearance of specific keywords.

For the first step, we used a keyword analysis tool developed at Lancaster, WMatrix [17]. WMatrix identifies relevant keywords by comparing word frequencies in a particular text against a corpus of standard English [3]. Because word occurrences are rare events [5], this comparison uses a specific test statistics, the log likehood, to assess the statistical significance of a particular word [5,18]. The technical details of this go beyond the scope of this paper, but in practice any log-likely hood higher than 16 indicates a significant keyword.

3. RESULTS

3.1 Data preparation

As the only preparation, we removed all bibliographies and reference sections from the articles to avoid conference names, which tend to show highly recurring patterns. This brought our sample corpus to 284,781 words. The word count by year of publication is shown on Table 1, and shows a bell-shape similar to the distribution of the article themselves (Figure 1).

1998	1999	2000	2001	2002	2003	2004
12,145	34,687	42,188	38,628	98,583	34,942	23,608

Table 1: Number of words per year of publication

Middleware (5153.9)
application (3484.6)
component (2593.4)
applications (2182.3)
distributed (2167.4)
components (2150.1)
object (2142.6)
interface (1681.7)
mobile (1627.8)
orb (1476.0)
reflective (1459.0)
adaptation (1404.4)
objects (1373.4)
architecture (1343.6)
dynamic (1336.8)
implementation (1320.8)
corba (1304.6)
systems (1284.2)
reflection (1250.3)
model (1187.4)
QoS (1169.8)
system (1134.5)
network (1014.3)
service (981.3)
resource (976.2)
configuration (958.0)
binding (957.0)
resources (928.1)
interfaces (907.8)
invocation (889.5)
code (843.8)
(can) (827.1)
management (824.5)
client (814.7)
framework (801.7)
context (801.1)
protocol (797.4)
(e.g.) (777.2)
support (746.3)
devices (742.5)
dynamically (737.1)
requirements (728.8)
(=) (718.0)}
communication (713.3)
(C) (710.5)
(such_as) (709.8)
services (703.4)

Figure 3: The 47 words with the highest log-likelihood in our top 40 article set.

keyword	rank	count	log-likelihood
Middleware	1	1683	5153.90
application	2	1313	3484.55
component	3	916	2593.36
reflective	11	491	1459.04
service	24	677	981.33

Table 2: Statistics for some of the top keywords of Figure 3

3.2 Keyword analysis

WMatrix returned 2053 keywords with a log-likelihood of 16 or higher. Of these most are related to Computer Science and Technology, but not specific to middleware research. However, the keywords with the highest log-likelihood (Figure 3, Table 2) tend to be middleware-related topics, which confirms the validity of the approach.

In this restricted list, what is surprising are the words that do *not* appear. For instance if we look at the challenges mentioned in a recent review of middleware research [10], some of the major issues mentioned (*Grid Computing*, *Ubiquitous Computing*, *heterogeneity* and *interoperability*, *coordination* and *scalability*) do not show up in Figure 3. In fact, except for 'open', they barely appear in the set of 40 papers (Table 3).

This could be explained by a combination of at least three reasons: (i) since these words denote challenges to be addressed, one can expect them to absent from past research; (ii) our set of article stops in 2004 and does not reflect more recent work; (iii) different communities have different vocabularies, and although 'ubiquitous' or 'pervasive' barely appear in our set of articles, closely related concerns such as 'context' (rank 36), and 'mobile' do (ranked 9) (Figure 3).

3.3 Keyword correlation

We selected the 47 top keywords returned by WMatrix, as we felt that words beyond this rank started to be less specific to middleware. Of these, we discarded 5 ('can', 'e.g.', '=',

keyword	rank	count	log-likelihood
open	190	310	245.45
wireless	222	82	210.23
heterogeneous	259	63	176.04
interoperability	301	52	159.24
embedded	405	51	120.7
ubiquitous	478	39	99.28
coordination	520	38	91.90
scalability	575	28	85.75
pervasive	1379	17	28.58
grid	2728	11	10.55
power	N/A	59	0.75
energy	N/A	12	0.19

Table 3: Statistics for some of the challenges covered in [10]

'C' and 'such as'). We also added the keywords shown on Table 3, to reflect the challenges of [10].

To avoid differentiating between different forms of the same word (singular/plural), or words closely related ('reflection' and 'reflective', 'mobile' and 'mobility', 'ubiquitous' and 'pervasive'), we furthermore grouped related keywords in categories using regular expressions (e.g. using 'architectur.*' to cover both 'architecture' and 'architectural'). In doing so we were careful not to overextend a category, e.g. preferring 'reflection | reflective' over 'reflect.*' as 'reflect' is often used for meta-discourse with no link to reflection.

Using the resulting 43 categories as a crib, we computed the number of occurrences of each category in each article. An excerpt of the results is shown in Table 4. We thus associated each keyword with a vector of counts (the columns of Table 4), that indicates the distribution of these keywords across the set of 40 articles.

ID	year	middleware	qos	adaptation	application	architecture	...	word count
1	1998	29	7	20	16	20	...	5464
2	1998	20	3	9	15	13	...	6681
3	1999	13	1	30	11	20	...	5149
4	1999	31	10	41	12	36	...	8014
5	1999	40	17	27	26	20	...	3822
6	1999	31	60	2	30	13	...	11224
7	1999	0	19	1	4	0	...	6478

Table 4: Word counts for some of our categories for the articles published in 1998 and 1999

Using these vectors one can compute a correlation coefficient for each pair of keywords that indicates whether they tend to appear together (positive correlation) or to repulse each other (negative correlation). A correlation coefficient is a number between -1 and 1 that indicates whether two series of numbers are linearly correlated. 0 indicates no linear correlation, 1 a perfect positive linear correlation and -1 a perfect negative linear correlation with a negative slope. As for word frequency, correlation coefficients need to be assessed for statistical significance: as a rule of thumb the smaller the sample population (as in our case) the less accurate coefficients are. We did not perform this important task in this early work. Instead we used correlation coefficients as indicators of *possible* links between keywords rather than irrefutable evidence thereof.

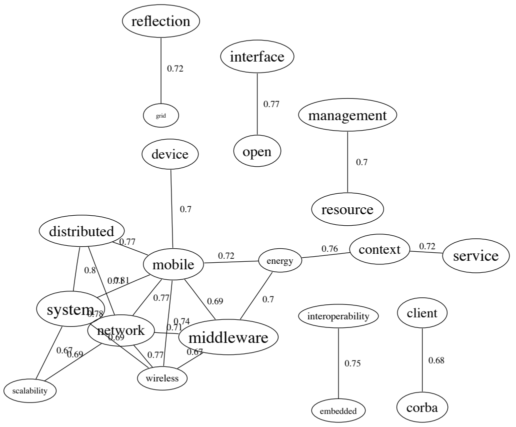

Figure 4: Positive correlations (≥0.65) between pairs of keyword categories

Of the $43 \times (43 - 1) \div 2 = 903$ pairs of keywords, 24 pairs had a correlation coefficient of 0.65 or more (Figure 4), and 7 pairs had a correlation coefficient of -0.28 or less (Figure 5). On these graphs, edges are labelled with the coefficient linking two keywords, and keywords are displayed in a font-size that reflect their frequency. The two cut-off values were chosen to represent a small set of the strongest positive and negative correlations.

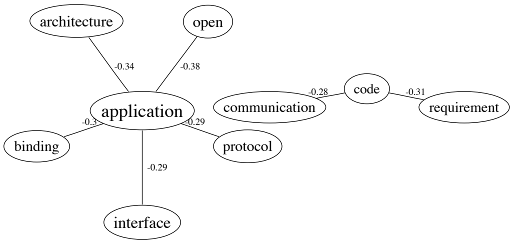

Figure 5: Negative correlations (≤-0.28) between pairs of keyword categories

As for the list of top keywords, some of the positive correlations seem obvious: 'distributed' and 'system', 'resource' and 'management', 'wireless' and 'mobile' are terms we naturally associate together. Others are understandable, but may point at potential research gaps: for instance, although interoperability is an issue in large-scale systems, it seems to have been preferably addressed in the context of embedded systems. Some correlations must be considered with care. The one between 'reflection' and 'grid' for instance is particularly weak, considering how little 'grid' appears in our corpus (15 times in 3 articles when counting hyphenated and plural terms).

Finally some pairs might be unexpected, such as the link between 'context' and 'service'. Figure 6 plots the number of occurrences of 'context' against the number of occurrences of 'service' in each article (counting both singular, plural and hyphenated expressions). This chart shows that all articles with a large number of 'context' mention 'service' a comparable number of times, and that there does indeed seem to be a linear trend between the two terms, usually a strong sign of some form of causality.

Contrary to what one might expect, 'service(s)' is here not linked to 'web service(s)'. 'web service(s)' only occurs 6 times in 3 articles, compared with 1262 for 'service(s)'. 'context' is however linked to 'context awareness' but not exclusively: the two-word expression appear 85 times in 7 articles (against 507 occurrences in 31 articles for 'context'), and the 7 article that contain both terms show a very strong correlation between the two (correlation coefficient 0.9036), a strong indication that these articles use 'context' to discuss 'context awareness'.

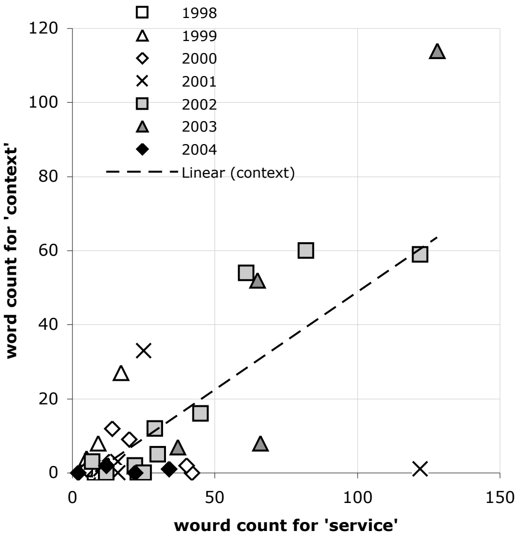

Figure 6: Occurrences of 'context' against occurrences of 'service' in each article, with years of publication.

The negative correlations (Figure 5), are more puzzling than the positive ones. The coefficients are, in general, much weaker, but if we look in detail at the strongest one, between 'application' and 'architecture', (Figure 7) (and other pairs show similar graphs), the two terms clearly tend to exclude each others: articles that discuss 'architecture' at length do not contain a lot of 'application' occurrences, and reciprocally.

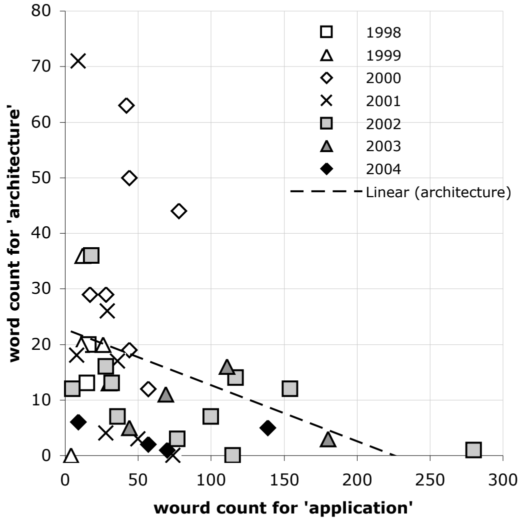

Figure 7: Occurrences of 'architecture' against occurrences of 'application' in each article, with years of publication.

3.4 Trends

Finally we analysed trends in the appearance or disappearance of keywords in our corpus. Because some years had more articles, we used average word densities for this analysis. Again, we computed correlations as a rough approach to spot interesting trends. Tables 5 and 6 list the keywords with the strongest positive and negative trends. For each keyword the tables indicates the correlation coefficient ("How linear is the trend?"), the average word density for the keyword, and the slope of the best linear fit ("How much word density does the keyword tend to lose or gain per year?"[2]). In addition, the temporal evolution of some of these keywords are shown on Figures 8 to 11.

word	correl.	avg. word density	slope
application	0.95	0.81%	0.172% p.a.
adaptation	0.70	0.38%	0.119% p.a.
service	0.61	0.44%	0.072% p.a.
client	0.66	0.14%	0.048% p.a.
code	0.67	0.12%	0.046% p.a.
network	0.79	0.22%	0.046% p.a.
system	0.65	0.68%	0.045% p.a.
interoperability	0.81	0.03%	0.010% p.a.
wireless	0.76	0.03%	0.009% p.a.
energy/power	0.75	0.03%	0.006% p.a.
embedded	0.55	0.02%	0.004% p.a.

Table 5: Keyword trends with the strongest positive correlations (≥0.55, 1998-2004)

word	correl.	avg. word density	slope
object	-0.60	0.54%	-0.177% p.a.
binding	-0.81	0.17%	-0.153% p.a.
interface	-0.73	0.35%	-0.136% p.a.
model	-0.79	0.36%	-0.094% p.a.
open	-0.84	0.24%	-0.090% p.a.
reflection	-0.72	0.33%	-0.066% p.a.
communication	-0.82	0.18%	-0.057% p.a.
architecture	-0.60	0.24%	-0.050% p.a.
implementation	-0.64	0.23%	-0.050% p.a.
protocol	-0.71	0.18%	-0.019% p.a.

Table 6: Keyword trends with the strongest negative correlations (≤-0.55, 1998-2004)

These numbers show some strong, and possibly unexpected trends. The use of 'application' and 'adaption' has risen substantially over the 6 years covered by our corpus (Table 5 and Figure 8). The rise of 'application' is particularly steady, with a regular and almost smooth increase.

From the keywords we selected to represent the challenges of [10], only four show up here: 'interoperability', 'wireless', 'energy/power', and 'embedded' (Table 5, and Figure 9 for the first three). Their evolution is much more volatile, which can be explained by their low level of occurrence in general, but Figure 9 definitely seem to show that they are on the rise.

The keywords with a negative trend (Table 6, Figure 10) contain some surprising members. Do 'object', 'reflection', 'implementation' denote concerns that are fading out? Or do these terms denote concepts that have been integrated so strongly in the reflective middleware community, that they

[2] The slope is expressed in word density (expressed as a percentage) per annum.

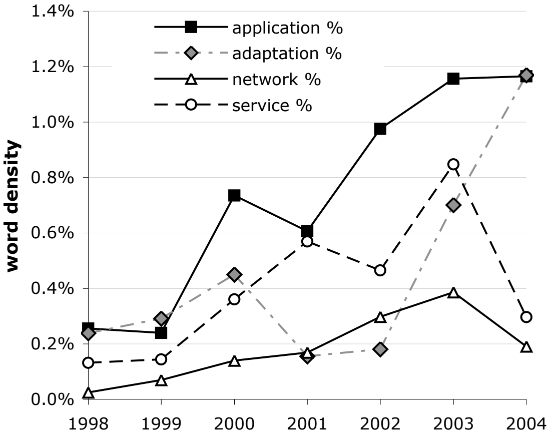

Figure 8: Some keywords with a general upward trend

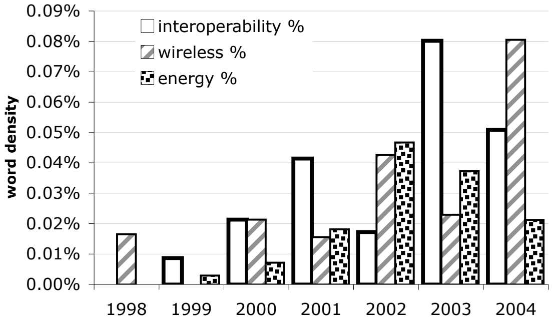

Figure 9: Keywords from [10] with an upward trend

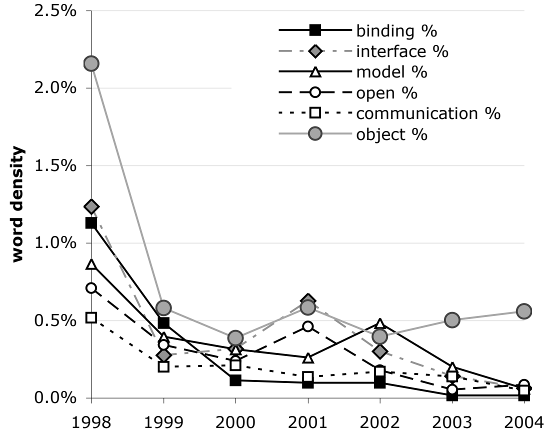

Figure 10: Some keywords with a general downward trend

do not need to be discussed any more? This certainly seem to be the case for 'object', as object-orientation or paradigms that evolved from it (components, frameworks) still underpins most of the middleware research, although object-oriented middleware itself might no longer be perceived as an active area of research.

Figure 11 focuses on of the evolution of 'architecture', and show how individual articles are distributed around the average density line. Along with Figure 8, this chart confirms the findings of Section 3.3: while the use of 'application' is growing, 'architecture' is fading out of fashion.

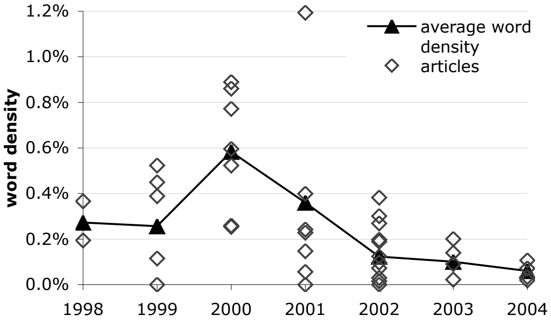

Figure 11: Evolution of the use of 'architecture'

4. DISCUSSION

4.1 Evaluation

The early results we have presented seem to show that keyword analysis can be a valuable tool to investigate the research in reflective middleware. Many of the correlations and trends we have uncovered confirm what many experts would say of the field. In that sense, they show the validity of the approach: it does return meaningful (albeit already known) results.

More interestingly, we have also uncovered unexpected links, such as the mutual repulsion of 'architecture' and 'application', the downward trend of 'architecture' (at least over the years we have considered), or the weak representation of the challenges identified by Issarny et al. in [10].

More in-depth investigations are needed to interpret these results. In that respect, this kind of statistical approach cannot replace an in-depth survey, but can guide and complement it. It sheds a new light on past research, can uncover unsuspected relationships and evolutions. Also, by prompting us to analyse surprising artifacts, it encourages us to ask new questions ("Is architecture irrelevant to middleware that target specific applications and if yes why?") It is also extremely scalable: computing word occurrences for the 47 categories we have used on 284,781 words only takes 1m46s on a 2GHz Intel Core Duo iMac.

4.2 Limitations and caveat

The use of corpus-based text analysis raises general issues of *scope*, *homogeneity* and *comparability* of the data, of *statistical reliability*, and of *interpretation* [18].

Scope, homogeneity and comparability One might question the set of articles we have selected. We have tried to aim for an homogeneous set, by choosing only published material from peer-reviewed venues, but others aspects are lacking. First, our data stops in 2004, which limits the significance of any trends or correlations to today's current research.

Second, by using by [2] as starting point, all articles are related to reflection by construction. We thus lack any baseline corresponding to 'general' middleware research. In that respect our findings do not reflect the impact of reflection on middleware research, but rather the internal evolution of the research related to reflective middleware.

Most importantly, one may criticise the way we selected our corpus: all articles on reflection and middleware do not cite [2]. Other articles that use similar but different techniques (such as aspects) may not either. These would need to be included in any in-depth study.

Statistical significance From a statistician's perspective our use of correlation coefficients is questionable at best. First, correlation coefficients are limited to detecting linear correlations. This might be good enough for keyword occurrences, as one might expect dependent keywords to either appear together or repulse each other, rather than showing more complex patterns. This approach is however definitely lacking for trends analysis, as it cannot detect 'bell shape' evolutions, where a keyword becomes popular before fading away.

In the same way that log-likelihood helps determine the significance of discrepancies in word-frequencies, the statistical significance of correlation coefficients must also be assessed. This usually takes the form of a confidence interval that reflects the precision of the coefficient value. We have not done this in this preliminary study, and our numerical results must therefore be always be interpreted by referring to the original data.

Interpretation Both because of the statistical nature of keyword analysis, and the somewhat limited number of articles we considered, our quantitative results are coarse and must be interpreted with care. They can guide further investigations, but are not sufficient to replace an in-depth analysis.

For instance we have found that 'reflection' is not correlated with any major keywords. Concluding that 'reflection' is not relevant to any of the topics we have identified would however be a gross misinterpretation: it is much more likely to be a pervasive theme, that does not tend to appear preferably along any other keyword.

5. RELATED WORK

Corpus linguistics has been applied to other areas of Computer Science, for instance to requirement engineering [19]. A number of approaches have also been proposed to quantify the impact of academic research, such as [15], which uses Social Network Analysis.

Probably one of the works closest to our aims is the in-depth impact analysis of middleware research presented by Emmerich et al. [7]. In this work, the authors propose the use of *impact traces* to document the influence of research on industrial products and practice. They rely on a large base of evidence: scholarly citations, the documents of standards bodies, the movement of people, interviews with experts. Although they do not look at reflection, their analysis is therefore much richer than ours but also requires a substantial effort. It might also be less applicable to reflective middleware, as in spite of early successes in open source products [8, 14], it remains a young technology. It was only proposed 10 years ago and Emmerich et al. observes latencies from 15 to 20 years between the publication of a key idea and its widespread use in industrial products. In fact, we see statistical text analysis as one of the possible sources on which impact traces could draw, to help the investigative process, but also uncover potential findings hidden to the human approach.

6. CONCLUSION

We have presented early results that show the potential of statistical text analysis for a better understanding of reflective middleware research. Although this approach does not replace a traditional survey work, it can complement it. Our early results show it can provide an overview of reflective middleware research from a slightly different viewpoint, and uncover unexpected trends and relationships.

A key benefit of this approach is its *scalability* as it can analyse a large number of artifacts at little cost. It is also quite *versatile*, and can be applied to any textual data. In particular we think that it could be applied beyond academic works to analyse how reflection has been adopted in production-grade projects (such a JBoss or ObjectWeb) by mining architectural documentation, e-mail discussions and even possibly source code.

7. REFERENCES

[1] G. S. Blair, G. Coulson, and P. Grace. Research directions in reflective middleware: the lancaster experience. In *ARM '04: 3rd workshop on Adaptive and reflective middleware*, pages 262–267, 2004.

[2] G. S. Blair, G. Coulson, P. Robin, and M. Papathomas. An architecture for next generation middleware. In *IFIP Int. Conf. on Dist. Sys. Platforms and Open Distributed Processing (Middleware'98)*, Sept. 1998.

[3] British National Corpus (BNC) Consortium. BNC Sampler Written Corpus. http://www.natcorp.ox.ac.uk/.

[4] L. Capra, W. Emmerich, and C. Mascolo. Middleware for mobile computing: Awareness vs. transparency. In *HOTOS '01: 8th Workshop on Hot Topics in Operating Sys.*, page 164, 2001.

[5] T. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[6] H. A. Duran-Limon, G. S. Blair, and G. Coulson. Adaptive resource management in middleware: A survey. *IEEE Dist. Sys. Online*, 5(7):1, 2004.

[7] W. Emmerich, M. Aoyama, and J. Sventek. The impact of research on middleware technology. *SIGSOFT Softw. Eng. Notes*, 32(1):21–46, 2007.

[8] M. Fleury and F. Reverbel. The JBoss extensible server. In *ACM/IFIP/USENIX Int. Middleware Conference (Middleware'03)*, pages 344–373, 2003.

[9] Google Inc. Google scholar beta. http://scholar.google.co.uk. accessed July 22 2008.

[10] V. Issarny, M. Caporuscio, and N. Georgantas. A perspective on the future of middleware-based software engineering. In *FOSE '07: Future of Software Engineering*, pages 244–258, 2007.

[11] F. Kon, F. Costa, G. Blair, and R. H. Campbell. The case for reflective middleware. *Commun. ACM*, 45(6):33–38.

[12] F. Kon, M. Román, P. Liu, J. Mao, T. Yamane, L. C. Magalhães, and R. H. Campbell. Monitoring, Security, and Dynamic Configuration with the dynamicTAO Reflective ORB. In *IFIP/ACM Int. Conf. on Dist. Sys. Platforms and Open Distributed Processing (Middleware'2000)*, pages 121–143, Apr. 2000.

[13] T. Ledoux. OpenCorba: A reflective open broker. In *2nd Int. Conf. on Meta-Level Architectures and Reflection, (Reflection'99)*, pages 197–214, 1999.

[14] OW2 Consortium. http://www.ow2.org. accessed on August 7 2008.

[15] G. L. Polites and R. T. Watson. The centrality and prestige of CACM. *Commun. ACM*, 51(1):95–100, 2008.

[16] P. Rayson. *Matrix: A statistical method and software tool for linguistic analysis through corpus comparison*. PhD thesis, Lancaster University, UK, 2003.

[17] P. Rayson. Wmatrix: a web-based corpus processing environment, computing department, lancaster university. http://ucrel.lancs.ac.uk/wmatrix/, 2008.

[18] P. Rayson, D. Berridge, and B. Francis. Extending the cochran rule for the comparison of word frequencies between corpora. In *In Volume II of Le poids des mots: Proceedings of the 7th Int. Conf. on Statistical analysis of textual data (JADT 2004)*, pages 926–936, 2004. ISBN 2-930344-50-4.

[19] P. Sawyer, P. Rayson, and K. Cosh. Shallow knowledge as an aid to deep understanding in early phase requirements engineering. *IEEE Trans. on Soft. Eng.*, 31(11):969–981, 2005.