Pacific Knowledge Systems

Experience with Long-Term Knowledge Acquisition

"Data from 100s of knowledge bases, show that it takes only a few minutes to add a rule, debug the knowledge base and have the system up and running again"

Paul Compton
School of Comp. Sci. & Eng.
University of New South Wales
+61 2 9385 6906
mkim978@cse.unsw.edu.au

Lindsay Peters, Timothy Lavers
Pacific Knowledge Systems Pty. Ltd
+61 2 9209 4852
l.peters, t.lavers@pks.com.au

Yang Sok Kim
School. of Comp. Sci. & Eng.
University of New South Wales
+61 2 9385 5644
yskim@cse.unsw.edu.au

Acknowledgements

The support of the Australian Research Council is gratefully acknowledged.

[1] Figure 12 and the error rate section of the paper differ from the ACM published version, because there was error in the calculations for Figure 12 in the published version. The changes do not alter the conclusions of the paper.

Abstract

Evaluation has remained a major challenge for knowledge acquisition and little data is available on how experts actually use knowledge acquisition technology. A number of companies offer Ripple-Down Rules to enable on-going knowledge acquisition and maintenance while a system is in use. One of these companies, Pacific Knowledge Systems has logged user activity over a number of years. Data from these logs demonstrate that domain experts continue to add knowledge to a knowledge base over years. The logs also demonstrate that new knowledge can be added very rapidly regardless of knowledge base size or age. We assume that the on-going knowledge acquisition observed was driven by the need to make changes and encouraged and allowed by the ease of the knowledge acquisition technology used. The question arises of whether experts in other domains would also chose to continue to add knowledge to their knowledge bases if this was supported.

Contents

Introduction

Knowledge acquisition research covers a wide range of topics, but one of the core research challenges has been to develop tools and methods for people to transfer their expertise to a knowledge base. Evaluation has always been a major challenge for this type of knowledge acquisition research, as the most appropriate forms of evaluation are studies of actual domain experts using knowledge acquisition tools and methods. In general this is too difficult both in terms of sufficient access to experts, or even “lay” experts, and appropriate controls in experimental design. The Sisyphus experiments proposed a series of knowledge acquisition tasks and provided relevant data and material for different research groups to build systems that could be compared [1]. The Sisyphus experiments produced very useful results as one could see how different tools and methods approached the same problem. However, this fell well short of true comparative evaluation as the systems were built from the materials provided by the knowledge acquisition research groups themselves rather than domain experts. Niche evaluation has also been carried out using simulated experts built by machine learning to provide “expertise” [2].

It is probably fair to say that most research on tools and methods for human knowledge acquisition has been forced to use persuasive arguments and one-off demonstrations. This is not a criticism; it is simply too costly to provide more empirical evaluation.

One issue arising from the difficulty of evaluation is that fundamental empirical questions about whether knowledge acquisition tools and methods are cognitively appropriate remain unexplored. The situated cognition debate concerned the central question of whether knowledge is something to be mined or extracted from the expert’s mind or whether knowledge is something that is constructed in a particular context [3] and necessarily incomplete [4]. This relates to the qualification problem[5] which states that it is impossible to anticipate all situations that will arise for a robot in the real world. Situated cognition similarly claims that the knowledge we express about the world is always partial and incomplete.

Ripple-Down Rules (RDR) were developed to enable knowledge to be continually added to a knowledge-based system while it is use. This does not solve the problems identified by situated cognition, but at least supports on-going fixes to the partial knowledge that humans provide. RDR have now been widely used for building knowledge-based systems while they are in use.

This paper presents summary data from the activity logs of 246 RDR knowledge-based systems, providing data on the number of cases processed, rules added, the time to add rules etc. This data cannot provide a proper comparative evaluation of Ripple-Down Rules, but it at least provides substantial empirical data on the use of knowledge acquisition tools which support on-going knowledge acquisition. Such data does not seem to be available for other knowledge acquisition tools.

[2] www.ivisgroup.com
[3] www.erudine.com
[4] http://domino.research.ibm.com/comm/research_projects.nsf/pages/caats.index.html
[5] www.pks.com.au

Ripple-Down Rules (RDR)

The essential feature of RDR is that rules are added to deal with specific cases (generally while the system is already in operation use). That is, the rule building process starts whenever a case is processed by the system and fails to give the correct output. New rules to give the correct output for the case are then added into the inference sequence. Rule placement is automatic and outside the control of the expert (or knowledge engineer). Since there is no knowledge structuring and no requirement to understand the knowledge base as a whole, rules can be added by domain experts very quickly and easily. There are now number of different RDR algorithms, SCRDR[6], MCRDR[2], NRDR[7] etc. It is beyond the scope of this paper to discuss them in detail, but they are all based on the idea of a fixed inference sequence so that a new or correction rule is added automatically into the inference sequence.

The strict inference sequencing in RDR enables a second type of knowledge acquisition support. An expert may add a rule for a case that is overly specific, but they can only introduce an error affecting the previous knowledge base by adding a rule that is too general, so that cases previously processed by the same sequence may now incorrectly fire the new rule. Such cases can be shown to the expert who either makes their rule more specific to exclude the cases, or accepts that the rule should apply to these cases. Suitable cases can be provided by saving the cases for which rules are added, known as 'cornerstone cases'.

The first RDR system in routine use using SCRDR inference was a 2000 rule system for chemical pathology[6]. The study here is based on Pacific Knowledge Systems proprietary RDR technology (RippleDown®) based on MCRDR[2] which allows multiple conclusions for a case. In an earlier report on use of this technology, chemical pathologists from one laboratory added about 16,000 rules across 20 knowledge bases over a 29 month period, at an average speed of 77 secs per rule [8]. This paper is based on the same type of logs of user activity, but from 246 knowledge bases from 17 different laboratory organizations, including both public and private laboratories and also including the laboratory reported in the previous study [8].

RDR are used in other areas apart from medicine. Ivis Pty. Ltd2, maintains the web site of TESCO the world's largest on-line grocer, using RDR [9]. Erudine Pty. Ltd3 (previously known as Rippledown Solutions), does not reference RDR but describes an identical approach in white papers e.g. for redeveloping legacy systems. Other companies offer RDR products as one of their range, e.g an RDR data cleansing product from IBM4 [10]. These examples are included to illustrate that RDR are used in other areas beyond medical diagnostic reporting which is the source of the log data described here.

Data Used

The data comes from logs of activity across 246 knowledge bases from 17 diagnostic laboratory customers of Pacific Knowledge Systems5 which have been used to process about 330 million sets of data or cases. These knowledge bases are used for providing detailed diagnostic comments on laboratory results across clinical chemistry, serology, immunology etc for reports to go out to referring clinicians. They are also used by some laboratories to manage workflow as well as to audit test requests advising whether the requests were appropriate given the patient history, payment rules and so. Laboratories tend to use multiple knowledge bases to cover different sub-domains. Sometimes the same expert will be responsible for knowledge building for different sub-domains, or there may be individual experts with more specialized responsibility. The typical procedure is that when laboratory reports are generated the test result data plus available patient history are passed to the knowledge-based system which generates a detailed interpretative comment, with perhaps multiple independent comments, to be added to the report. The report is validated by senior domain experts, generally pathologists, and if the comment is inappropriate or missing, an appropriate comment will be added, the report will be sent to the referring clinician in the normal way and the report with appropriate comment will be referred for knowledge building. Not all reports will be checked as the pathologist expert may decide after some time that some reports only need to be looked at periodically or under certain conditions, and so allow some level of auto-validation. This corresponds to the normal practice of testing and evaluating a system before it is put into routine use, but here the system is monitored while it is in use, and the level of monitoring can be adjusted depending on how the system has performed.

New rules can then be added for cases for which comments were changed by the validating expert. The person adding the rules may or may not be the same expert who carried out the validation. The system records the time from when the expert calls up the case to when they have finished adding rules and exit the knowledge-builder module. Since the case already has a new comment, their task is to select features in the case which justify the comment. These features are used as the rule conditions. The rule is then verified against cornerstone cases, and if necessary further features are selected to refine the rule. Because composite comments are supported, the expert may add more than one rule for a case. The normal domain task of assigning an appropriate comment to a report was carried out at the report validation stage. The time logged for the rule-building sessions covers the knowledge engineering tasks of selecting rule conditions, which may include developing functions using the rich function language provided, and verifying and adjusting the resultant rules. Since there is no other knowledge engineering task in adding rules, the logged time represents the maximum time the expert spends on knowledge engineering. There is nothing that can be done away from the knowledge builder module except the normal domain expert task of writing an appropriate comment, which the expert should do anyway.

Since the logged time covers the time from when a case is called up for knowledge building until knowledge building is complete it includes any waiting period that the user experiences before the next cornerstone case is presented to them. It also includes time on extraneous activities such as answering the phone, getting coffee, or simply walking away from the computer screen for a prolonged period. We therefore use the median rather than the average time in the discussion below. The logs record the total number of cases for which rules are added in the month and the total time, so more exactly we use the median of the monthly average. The average of the monthly average is about twice the median, but shows the same relation to knowledge base size and age. The logs do not provide information on how many rules are added for a case, only the overall knowledge-building time. We assume that only one rule is added for most cases.

In the following discussion we use the term knowledge acquisition event (KA event) interchangeably with the term rule. We only have log data on KA events, not on rules added or total rules in a knowledge base. The number of KA events may underestimate, but cannot overestimate the number of rules, because there is at least one rule for each KA event. Any underestimate of knowledge base size in the following strengthens rather than weakens the results of the study.

The KA events only cover knowledge building while the knowledge base is in use. If rules were added prior to the knowledge base being putting into use, this is not captured. Some knowledge bases may also have their name changed and may appear to be a new knowledge base, with only new KA events counted under the new name. It would be possible to resolve such questions by obtaining detailed information about the 246 knowledge bases from the laboratories but this was beyond the scope of this study.

Results

The logs cover 246 knowledge bases and 183 of the 246 knowledge bases were in routine use in 2010 providing diagnostic comments for patient reports or auditing whether test requests were appropriate, and medical alerts. Logs from Pacific Knowledge Systems non-medical customers are not included. The 63 knowledge bases not in use in 2010 were all developed by laboratories who had multiple other knowledge bases in use in 2010 and were continuing knowledge building, so they do not represent abandoning the technology, but it was beyond the scope of this study to investigate the history of individual knowledge bases.

Knowledge Base Sizes

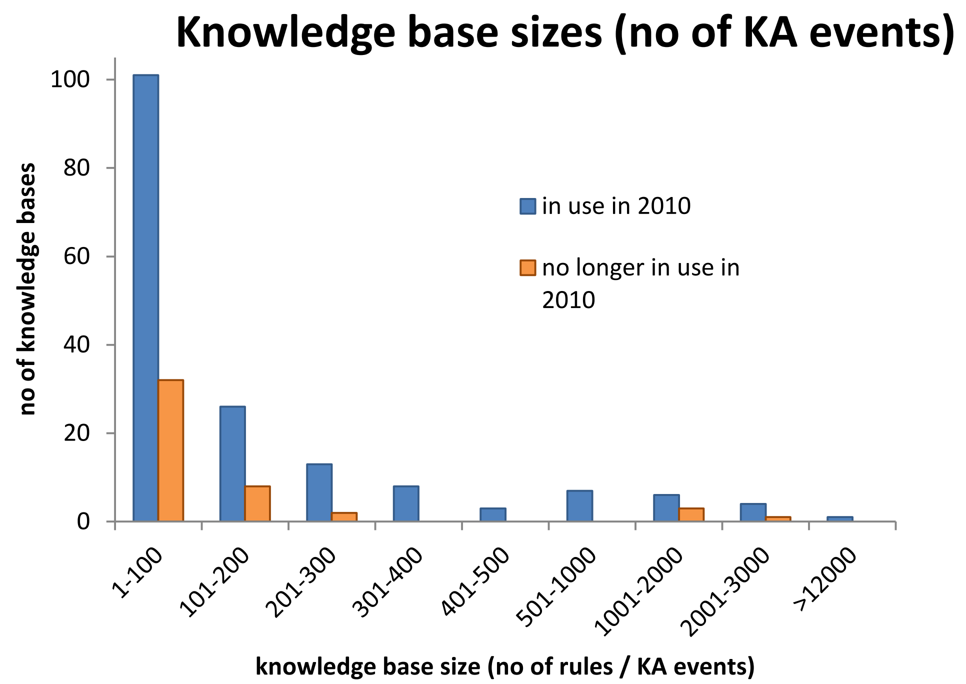

Figure 1 shows the number of knowledge bases of various sizes.

These are the final sizes in the log data. The size of the knowledge bases, both those in use and 2010 and those not in use (at least not under the original name) are shown in Figure 1. The majority of the knowledge bases are small, but eleven have had over 1000 KA events, so have at least 1000 rules and one of these has at least 12,737 rules.

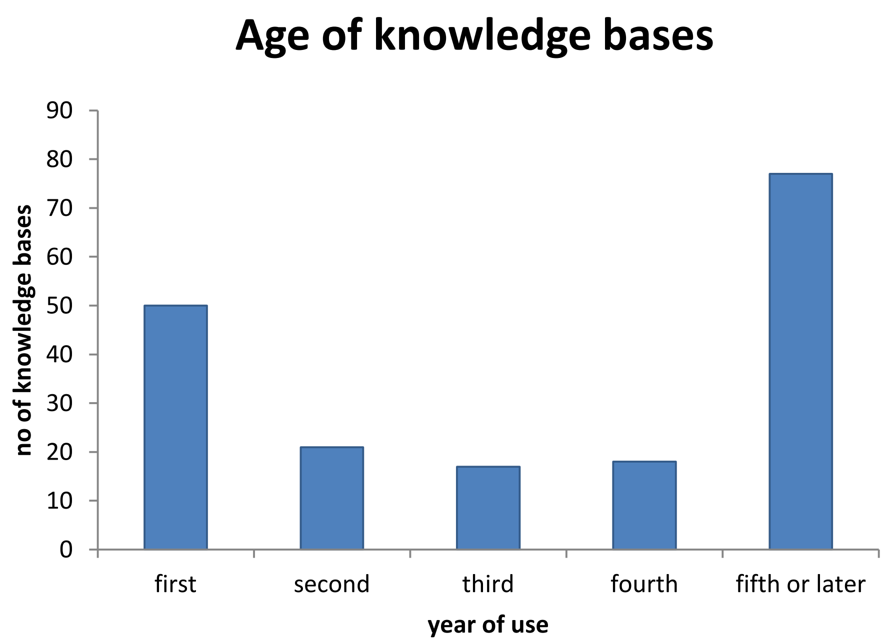

Figure 2 shows the length of time the knowledge bases in use in 2010 have been in use.

50 of the 183 knowledge bases have been in use for a year or less, but over 70 have been in use for five or more years.

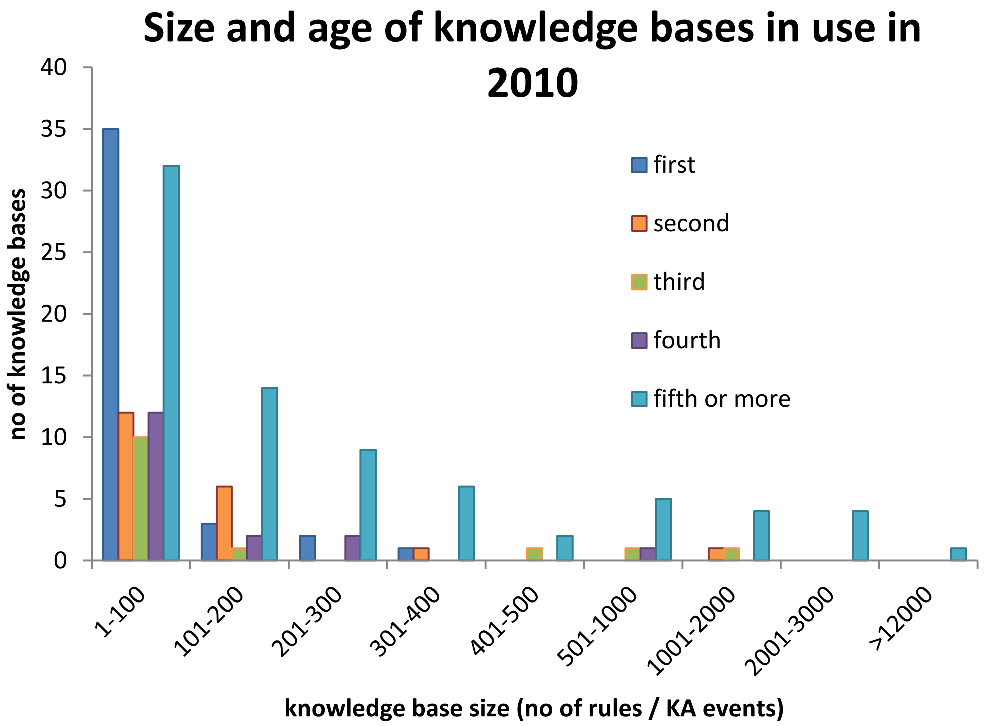

Figure 3 shows the same information related to knowledge base size (i.e. number of KA events).

For each knowledge base size it shows whether the knowledge bases were in their first year of use, second, third, fourth or had been in use for 5 or more years in 2010. As might be expected, the larger knowledge bases tend to have been in use longer. Of the smallest knowledge bases, with under 100 KA events, over 30% are in their first year as would be expected, but another 30% of these knowledge bases have been in use for five or more years.

There are a number of factors that determine knowledge base size. The most obvious factor is the size of the domain, as it is up to the laboratories to decide how many individual knowledge bases they wish to develop and the coverage of individual knowledge bases (e.g. separate knowledge bases for diabetes and thyroid comments). Figure 4 illustrates that the single knowledge base with 12,737 rules processes the largest number of cases. This may be because this a particularly high-volume domain, or perhaps because it covers a wider range of chemical pathology; however it is a single data point and in contrast the four next largest knowledge bases (2001-3000) on average processed the smallest number of cases. Overall there does not seem to be a relationship between knowledge base size and number of cases processed. Across all knowledge bases an average 82,571 cases were processed per knowledge base per month in 2010.

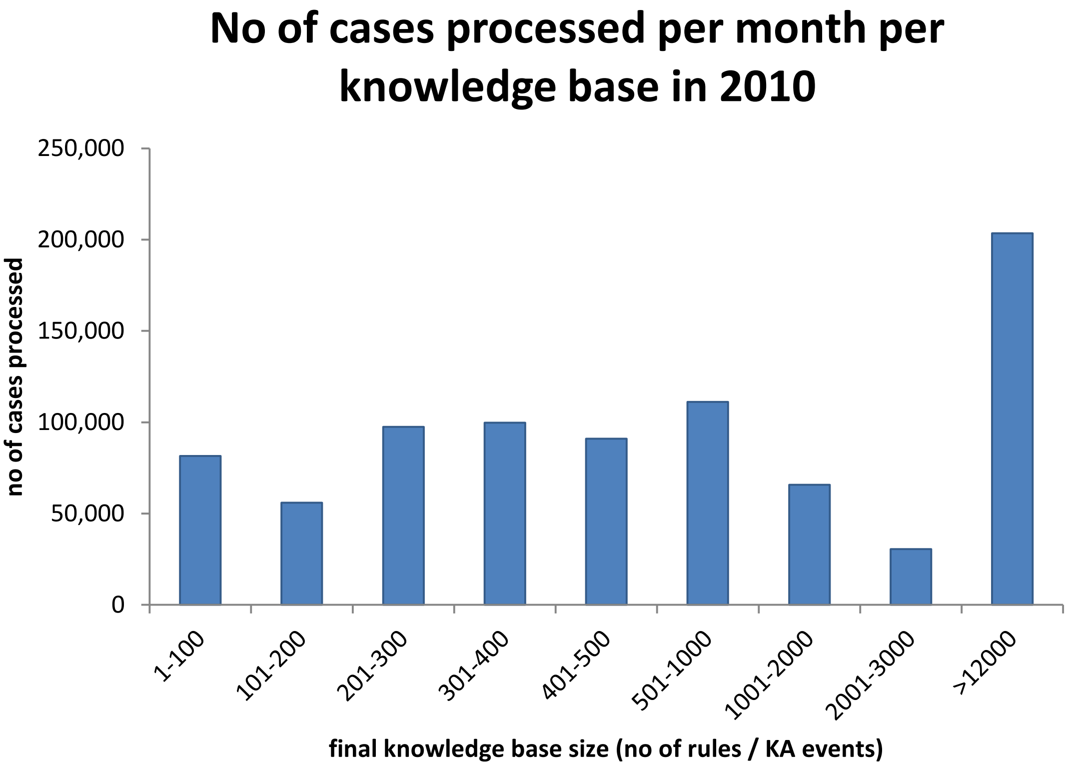

Figure 4 shows the number of cases processed per month per knowledge base in 2010 related to knowledge base size.

Although the size of the knowledge base will relate to the complexity of the domain, domain complexity is determined not just by the domain but also by the subtlety of the comments that the experts choose to make. For example if a report includes an elevated cholesterol, a very simple comment might be made that the cholesterol is elevated. However, the pathologist may also take into the patient's previous records and current treatment and if appropriate, may make a comment along the lines: "Raised cholesterol level persists on Zocor treatment. Consider increasing dose of Zocor and repeat lipid profile in 4 weeks.

Note that hypothyroidism may impair response to Zocor; suggest TSH level at time of next review." Clearly a comment like this is potentially more helpful to a GP, particularly in reminding the GP that thyroid disease may affect lipids, but making such a comment depends on the system having access to information about treatment, previous cholesterol results, and that there were no previous TSH results for the patient, as well as the current cholesterol results. It also depends on the domain expert taking the initiative to write the more sophisticated rules required. In the highly complex Bone Mineral Density domain each report comprises up to 50 comments. In this case, the expert is motivated to provide such a detailed report because referring GPs do not readily understand the raw results in this specialised domain. Although this is a low volume domain there is a significant time saving in an automated report compared to writing or dictating the report, war- ranting the knowledge acquisition effort involved.

A further issue is that although a knowledge engineer to structure the knowledge base is not needed, the pathologist needs to develop some knowledge engineering skills, not only in using the range of functions provided, but they also need some insight into the appropriate granularity of rules. If the pathologist creates rules that are inappropriately general, unnecessary refinement will be needed. On the other hand highly specific rules will mean that more rules will have to be added to cover the domain.

The key feature of RDR is that it enables rules to be added after the system is already in use. Figure 5 shows the number of knowledge bases to which rules have been added during each year of use. It also shows the year of use.

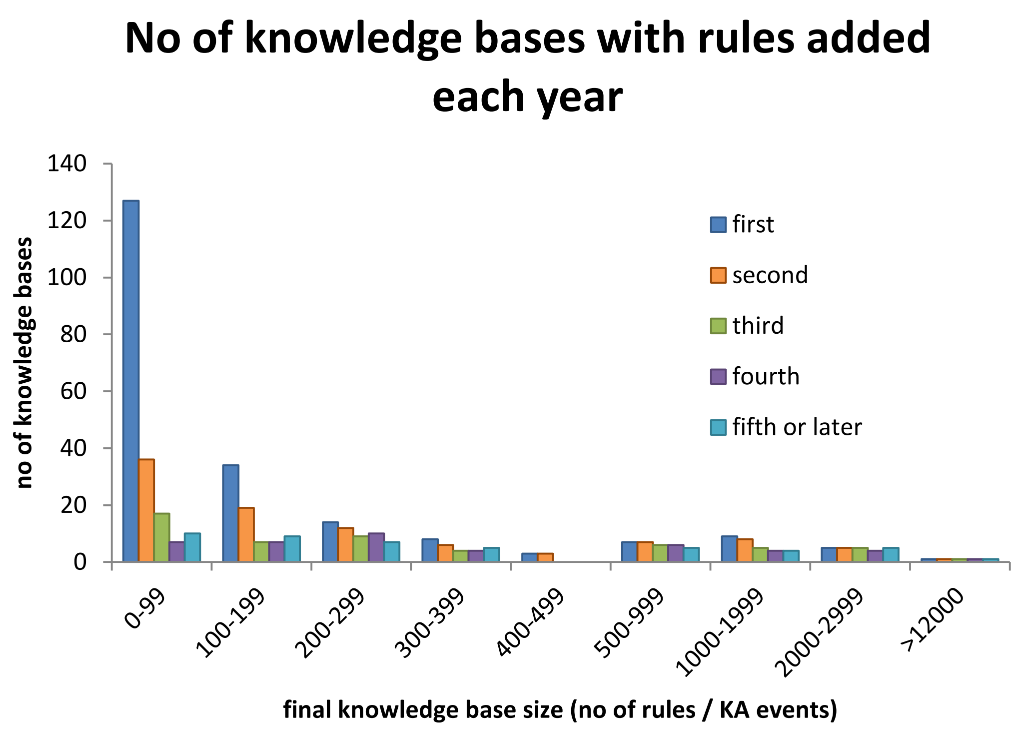

Figure 5 shows the number of knowledge bases to which rules have been added during each year of use: first, second, etc. This figure shows results for all 246 knowledge bases, not just the 183 in clinical use in 2010.

The same data in figure 5 is shown in figure 6 as a percentage of the number of knowledge bases of that size. The most remarkable feature of this data is that knowledge bases have rules added in later years and that this occurs even with small knowledge bases. 27% of the knowledge bases with less than 100 rules (KA events) have rules added in the second year and 13% have rules added in the third year. For knowledge bases of 200 to 300 rules, 80% have rules added in the second year, and five years and beyond, 47% are still having rules added.

The knowledge bases in figures 5 and 6 may have had only one rule added each year after the first year, so figure 7 provides data on the total number of rules added after a knowledge base has been in use for at least a year. Again even for knowledge bases of less that 100 rules 10% of the KA events occurred after a year in use. In general the larger the knowledge base, the more rules are added later, except for the three 400-500 rule knowledge bases.

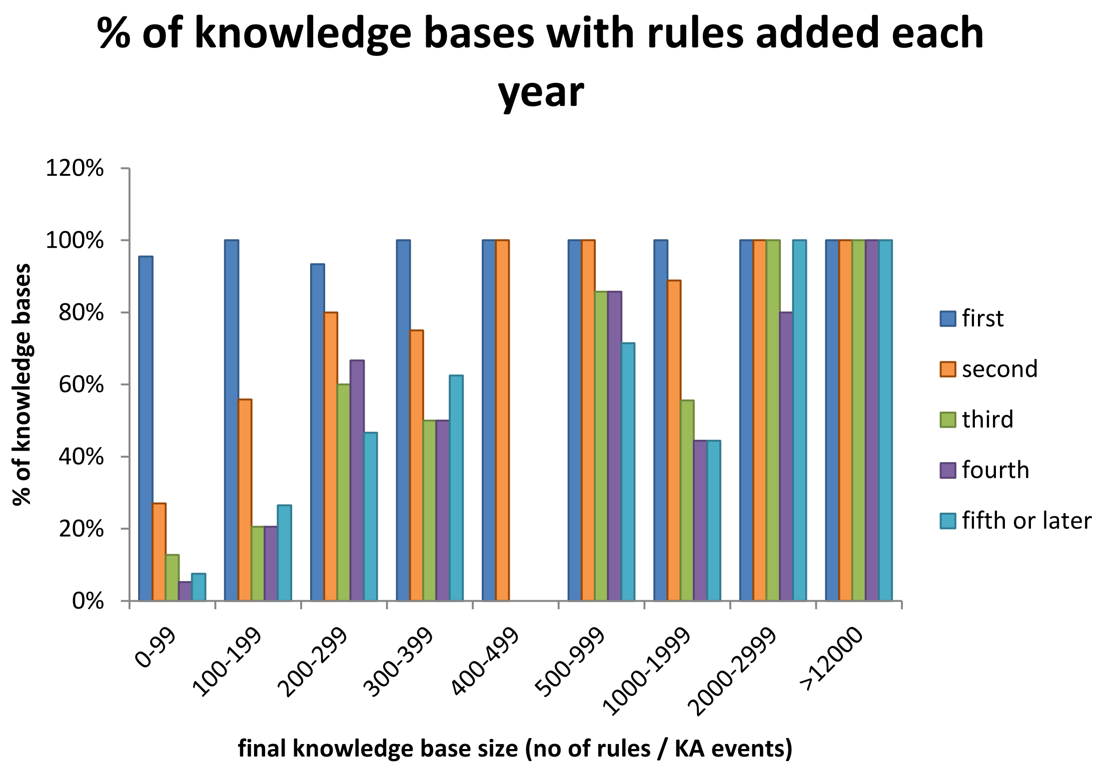

Figure 6 shows the same data as in figure 5, but expressed as a proportion of the number of knowledge bases.

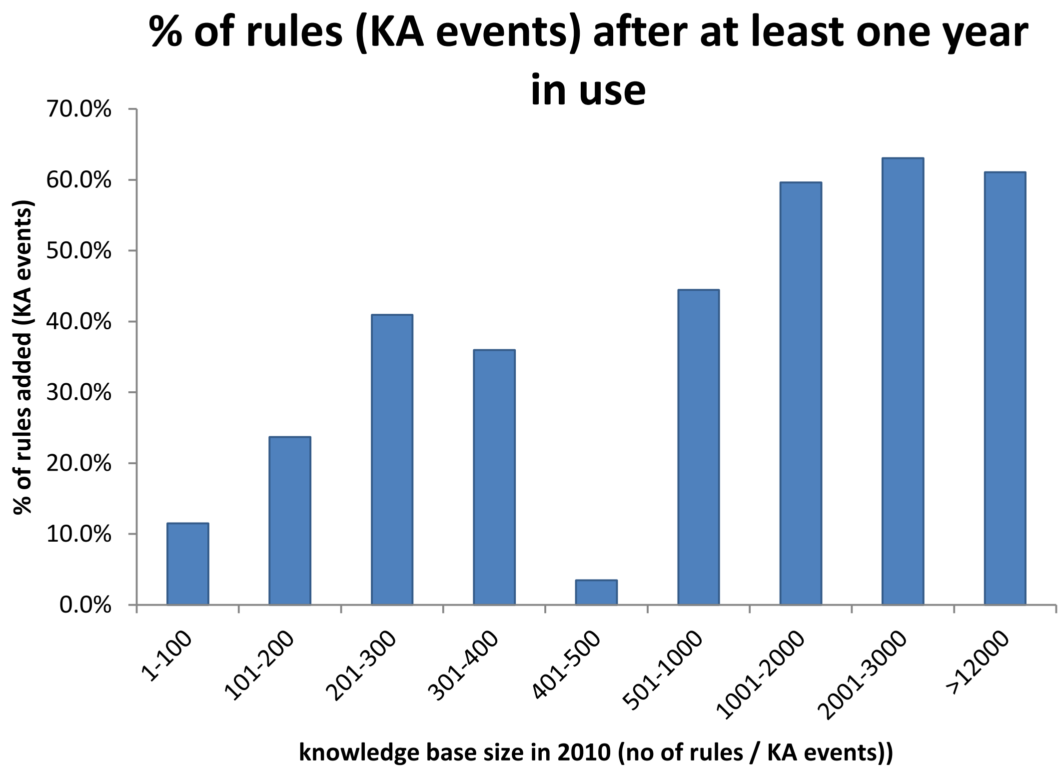

Figure 7 shows the number of KA events after at least one year in use. It shows the number of KA events after a year in use as a proportion of the total number of KA events for knowledge bases of a particular size. It only covers knowledge bases in use in 2010.

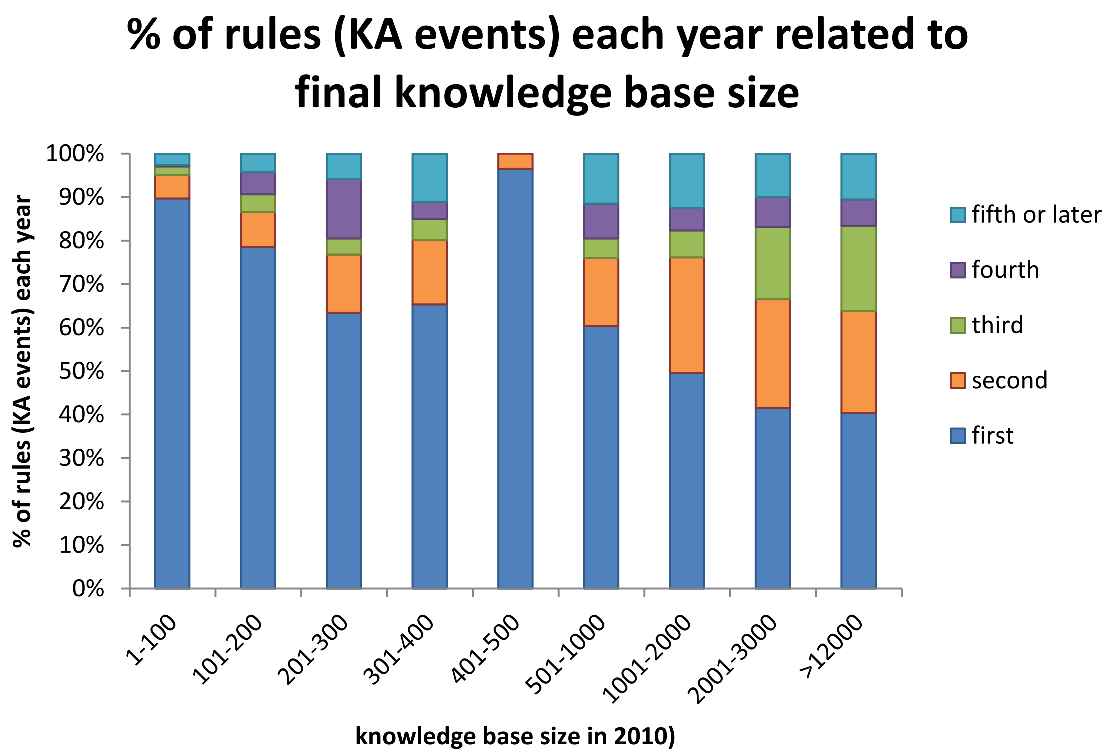

Figure 8 shows the rules added each year as a proportion of total rules for different knowledge base sizes. It only covers knowledge bases in use in 2010.

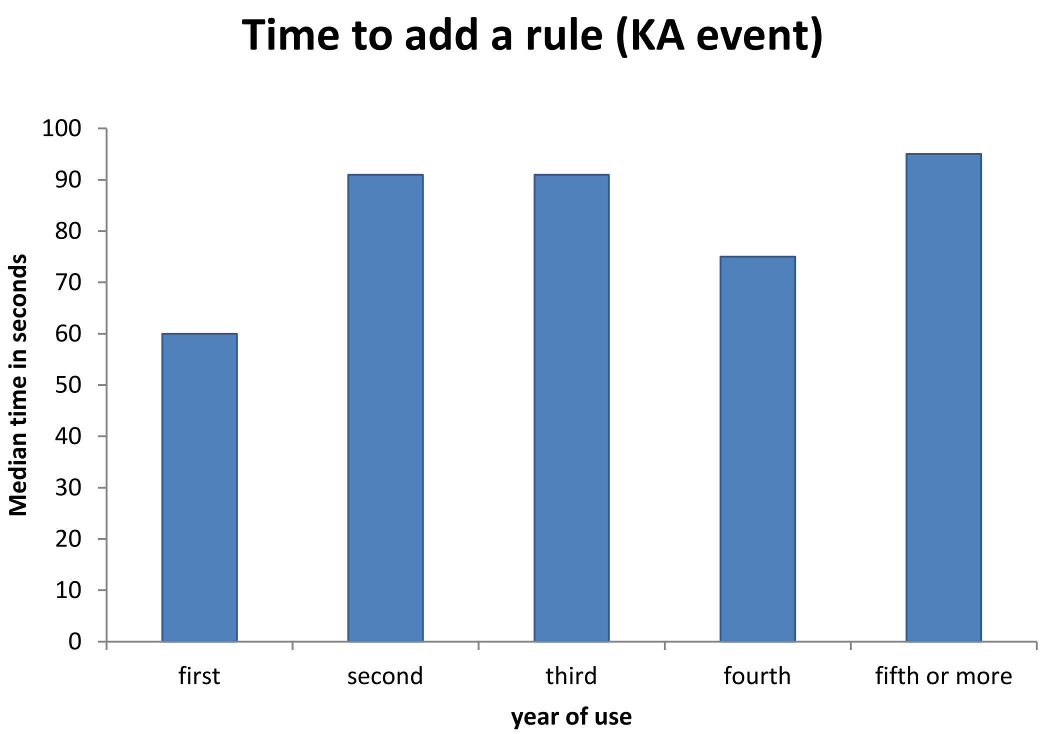

Figure 9 shows the median time to add a rule across all knowledge bases related to the year in use. This data is from all 246 knowledge bases.

Figure 8 shows the same data as for figure 6, but as a proportion of the total rules added (or KA events). As would be expected more rules are added in the first year than any other year, with more than 50% added in the first year for knowledge bases less than 1000 rules and even for the largest knowledge bases more than 40% are added in first year.

There appears to be little data on the need for on-going knowledge acquisition or maintenance for deployed knowledge-based systems, apart from the landmark study on XCON, but in this the growth in knowledge was explicitly due to the introduction of new computers [11]. The data here suggests on-going knowledge acquisition, if supported by the tool, is a very significant activity more generally. We will discuss whether this represents a failure of the RDR approach, but clearly experts are willing to keep on adding knowledge.

Time Required for Knowledge Acquisition

The median time to add a rule is shown in Figure 9. As discussed, medians are used because there is no control over whether the log data includes time where the expert was away from the computer screen engaged in other unrelated tasks. The times are remarkably short with about 60 seconds as the median time to add a rule in the first year and about 90 secs for most later years. The reason the time to add a rule is so short is that the task is so simple and there is no requirement to understand the knowledge base as a whole or its overall structure. If an expert decides that a case needs a new or better comment, they have already identified the features in the data that lead to this comment – otherwise they would not make the comment. This is not a knowledge engineering task, it is the normal pathologist task of looking at a report and deciding that it would be useful to provide a comment about the results for the referring GP. However, once this normal domain task is done, the knowledge engineering task of simply identifying the same features on the knowledge building screen, and perhaps some features to exclude cornerstone cases, is very simple. It will not be so simple if the expert wishes to use or formulate some complex function to represent the feature they have observed in the data. Unfortunately we have no data on individual rule building; however, the median data shows that generally the time to build a rule is extraordinarily small. As noted, the time logged is the time to build all the rules required for a case so that the median time for a single rule may be less than the times shown.

The main factor that might cause an increase in knowledge acquisition time is likely to be the number of cornerstone cases that have to be reviewed in verifying that the rule is appropriate. Anecdotally it has always seemed that experts only need to consider two or three cases before they have a sufficiently precise rule to exclude all the other cases that might fire the rule inappropriately but this has not been verified. Figure 10 shows the time to add a rule in the first year of use against knowledge-base size. We show data for just the first year to try to distinguish effects related to the size of the knowledge base from knowledge base age and perhaps decreasing familiarity. Knowledge bases have not grown to more than 2000 rules within the first year but for knowledge bases of less than this size, there is an increase in the median time with knowledge base size. However, a median time of 156 seconds to add a rule to knowledge bases between 1000 and 2000 rules is still very small.

[6] Since this paper was first submitted, this laboratory's computer platform has been upgraded and the expert has commented that there is 10 fold decrease in processing time.

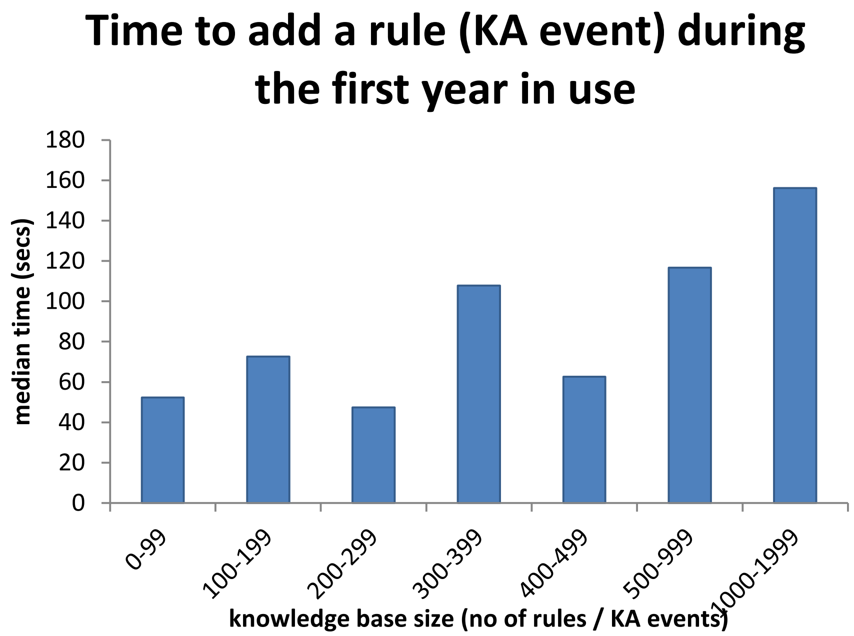

Figure 10 shows the median time to add a rule in the first year of use for knowledge bases of various sizes. This includes all 246 knowledge bases.

Figure 11 shows the median time to add a rule related to knowledge base size, ignoring the age of the knowledge base. The last three columns come only from the 12,737 rule knowledge base. Even though this is a single knowledge base it is clear that the time to add a rule(s) for a case increases significantly with knowledge base size. In this instance we did contact the domain expert who thought that his task remained the same as with other knowledge bases he maintained, but that he would tend to do something else while the system collected and processed the cornerstone cases, so the increased time was not because of an increased expert task6. At this stage, we have to take the results for this knowledge base on face value that above 3000 rules the logged time to add a rule rapidly increases with knowledge based size. However, even above 10,000 rules the median time to add a rule is only 607 seconds – well short of the usual expectation of how long it takes to add a rule to a very large knowledge base. Overall, across all knowledge bases the median of the average time each month for a KA event is 78secs. Alternately the average of the average including all data is 136 secs.

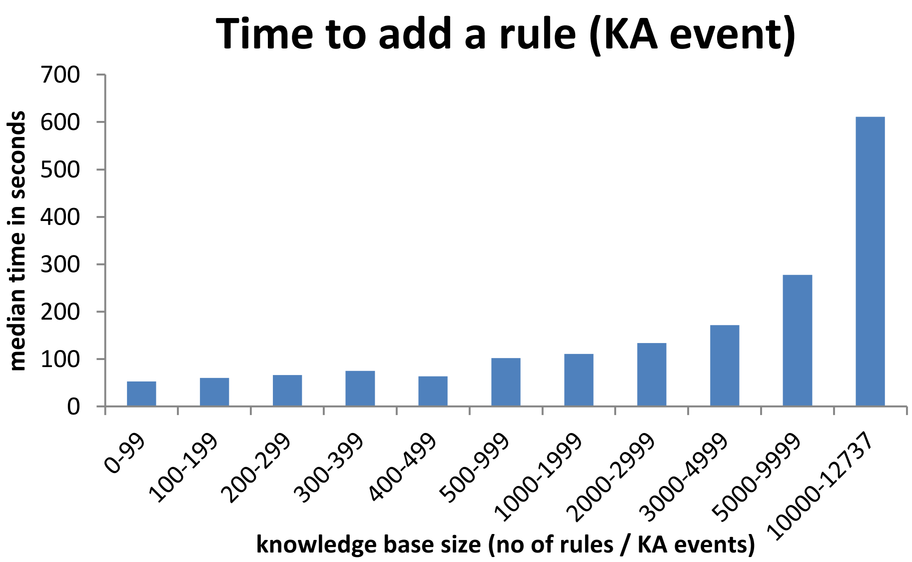

Figure 11 shows the median time for a KA event across all 246 knowledge bases related to knowledge base size at the time of rule acqusition. The last three columns all come from the 12,737 rule knowledge base.

Error Rates

The critical question in this is why experts chose to keep adding rules. The obvious answer is that they are dissatisfied with the performance of the knowledge-based system and keep trying to correct errors. The logs contain very reliable data on errors, because after reports have comments attached they must go through the validation stage and unless a comment is specifically rejected at this stage the report will go out with that comment.

The rejection rate therefore measures the rate at which comments are rejected; there is no other way to reject or change a comment. Experts may not look at every single report, but this only occurs because they decide the quality of certain comments is such that they can set some level of auto-validation for those particular comments. So the overall the rejection rate reflects the experts view of what comments are appropriate and what are not appropriate or should be improved.

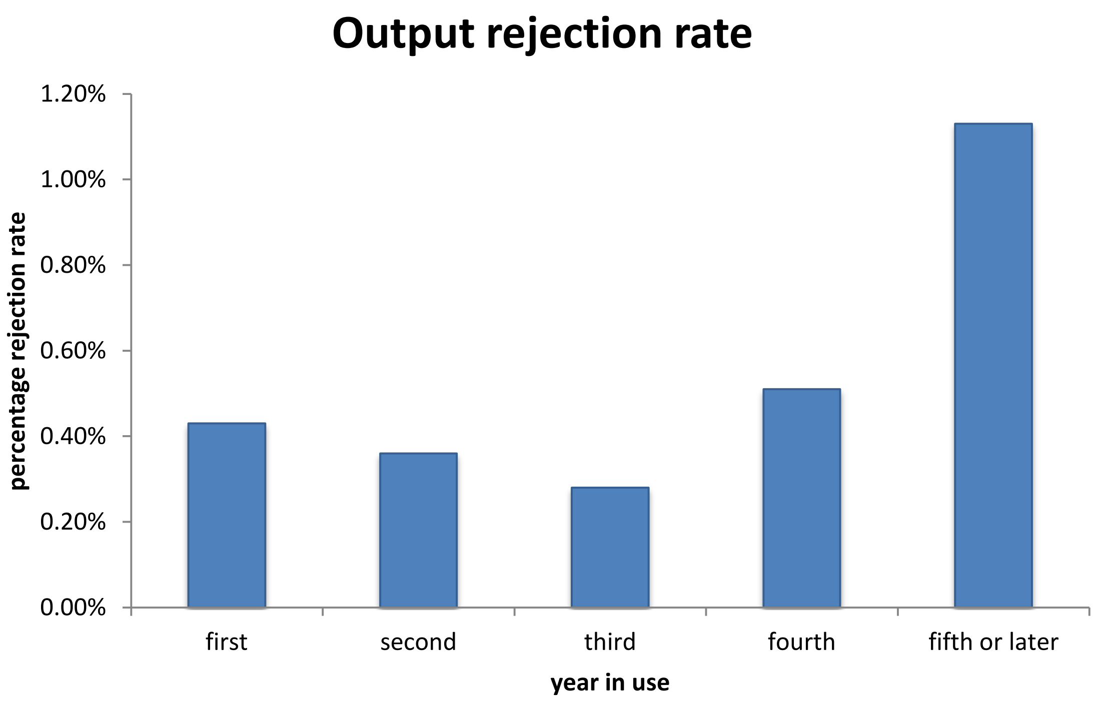

Figure 12 shows the rate at which the comments output by the 246 knowledge based systems are rejected. [2]

The data is Figure 12 overall average rejection rate, the total number of cases rejected divided by the total number of case processed by all 246 knowledge bases. One might expect the rejection rate in the first year to be higher than later years. Figure 12 shows that the first year rejection rate of 0.43% drops to 0.28% by year three, but increases thereafter. As discussed below, later increases in rejection may be due to changes in commenting, but there is no consistent pattern across the knowledge bases.

There is no notion of an independent 'gold standard' for the correctness of conclusions in this approach; here the domain expert is the 'gold standard' and whatever conclusion the expert chooses is by definition the correct conclusion.

[2] Figure 12 and the error rate section of the paper differ from the ACM published version, because there was error in the calculations for Figure 12 in the published version. The changes do not alter the conclusions of the paper.

Compared to typical error data for knowledge based systems, these rejection rates are very low with the overall average rejection rate being 0.83%. This cannot be because experts are ignoring errors; there can be very serious implications for inappropriate medical reports and accuracy is a major concern for all pathologists. We suspect that what happens is that experts very rapidly add and correct rules where an error would be significant, with perhaps many of these added off-line, but then over time they include more 'value-add' rules to assist the referring GP, for example the earlier comment about elevated cholesterol. Similarly subdomains may be added to a knowledge base incrementally, so that there is flurry of error correction when a new subdomain is added. Another important issue is that medical practice evolves, so that what might be the most appropriate advice changes. For example it was accepted practice to vigorously treat type 2 diabetes. Recent studies, revealed a higher risk of death in patients following vigorous treatment and the current practice is to ease off. So a comment on a low haemoglobin A1c level is likely to be one of caution rather than approbation [11]. Similarly rules for prescription of drugs such as lipid lowering drugs, related to pharmaceutical benefit schemes, periodically change, perhaps leading to different comments. In summary, much of the advice provided by these pathology knowledge-based systems falls into the category of highly useful patient specific management advice given current best practice, rather than simple diagnosis.

It should also be noted that overall about three times as many reports were rejected at validation as there were KA events. Some of this difference is because there may be a delay of a few days between a report being rejected and the designated knowledge builder adding the correction rule. During that time the incorrect comment may be given and rejected several times. Secondly, a different expert may be responsible for rule building than for validation and the rule building expert may decide not to add a rule for that comment. Finally however, much of the difference is because laboratories also use the knowledge bases for internal quality control. That is, a comment might advise that an analyser be checked. Such a comment will always be rejected, so that it does not go out to the GP, but the comment is useful and does not need to be corrected with a further rule.

Discussion

The data presented show unequivocally that RDR allow for very rapid knowledge acquisition. The total time to add more than 57,626 rules was about one man-year – and this time is an overestimate because the logs include interruptions to knowledge building. Secondly, although the time to add knowledge increases with large knowledge bases it remains extremely short – at about 10 minutes per rule or KA event. It seems likely that the increase in time is due to processing and data retrieval rather than an increased demand on the expert, but this needs further verification. Across all the rules added the median or average time is about 1-2 minutes per rule or KA event.

Probably the most interesting aspect of these results is the ongoing knowledge acquisition that is revealed. 240 of the 246 knowledge bases had rules added after they had been in use for more than a year. That is, for almost every knowledge base the domain expert decided more knowledge should be added after more than a year of use. In total at least 45% of the 57,626 KA events, while the systems were in use, with each event adding one or more rules, occurred after the systems had been in use for more than a year. Since there is little interest in on-going knowledge acquisition in the literature, we assume these figures would be seen as surprisingly large.

This on-going knowledge acquisition is unlikely to be due to an intrinsic limitation of RDR, so that there is repetition in the knowledge added. Firstly, there has been considerable earlier research showing that RDR, particularly MCRDR-based systems, learn efficiently e.g.[2]. Secondly, in the data here the rate at which experts reject cases and correct errors is very low. Thirdly, if there were repeat corrections being made, one would have expected some negative comments from the experts.

Both situated cognition [3] and McCarthy's qualification problem [5] suggest the likelihood of on-going knowledge acquisition. Experts do not and cannot anticipate all possible contexts in which knowledge will be used, so on-going knowledge acquisition is to be expected, with new contexts emerging as more cases are processed. If it is sufficiently easy to deal with new contexts as they appear, on-going knowledge acquisition would be expected.

We suspect the major reason for the scale of on-going knowledge acquisition observed, is simply because it is possible, so that experts with only a very small extension to their normal duties, over time can improve the quality of the advice they give – and the service they provide. We hope to investigate this hypothesis, as well as the many other questions that arise from this data, by investigation of the individual knowledge bases and interviews with the experts involved, but this has been beyond the scope of the present study.

As far as we know these medical diagnostic laboratory domains are the only domains where substantial data is available on the use of knowledge acquisition technology on an on-going basis. Would the same behaviour arise in other domains if the technology used supported very easy on-going knowledge acquisition? Do the data here illustrate a general feature about human expertise? Does this relate to what might be able to be achieved with ontologies [13]? Do the results here suggest that people might always tend to maximize the possibilities for individual expression, even in a controlled environment where any differentiation has to be justified?

References

[1] Gaines, B., Birmingham, W. P. and Schrieber, A. T. e. Sisyphus II Special Issue. 1996. *Journal of Human-Computer Studies*, 44(3-4).

[2] Kang, B., Compton, P. Lee, K. and Preston, P. 1998. Simulated Expert Evaluation of Multiple Classification Ripple Down Rules. In *Proceedings of the 11th Banff Knowledge Acquisition Workshop.* http://ksi.cpsc.ucalgary.ca/KAW/KAW98/KAW98Proc.html

[3] Clancey, W. J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations (Learning in Doing - Social, Cognitive and Computational Perspectives)*. Cambridge University Press.

[4] Compton, P. and Jansen, R. 1990. A philosophical basis for knowledge acquisition. *Knowledge acquisition* 2: 241-257.

[5] McCarthy, J. (1977. Epistemological problems of artificial intelligence. In *Proceedings of the 1977 International Joint Conference on Artificial Intelligence,* pp1038-1044.

[6] Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Menzies, T., Preston, P., Srinivasan, A. and Sammut, C. 1991. Ripple down rules: possibilities and limitations. In *Proceedings of the 6th Banff Knowledge Acquisition Workshop.*

[7] Beydoun, G. and Hoffman, A. 1998. Building problem solvers based on search control knowledge. In *Proceedings of the 1998 Banff Knowledge Acquisition Workshop.* http://ksi.cpsc.ucalgary.ca/KAW/KAW98/KAW98Proc.html

[8] Compton, P., Peters, L., Edwards, G. and Lavers, T. G. 2006. Experience with Ripple-Down Rules. *Knowledge-Based System Journal*, 19(5): 356-362.

[9] Sarraf, Q. and Ellis, G. 2006. Business Rules in Retail: The Tesco.com Story. *Business Rules Journal*, 7(6) http://www.BRCommunity.com/a2006/n2014.html.

[10] Dani, M., Faruquie, T., Garg, R. and Kothari, G. 2010. A Knowledge Acquisition Method for Improving Data Quality in Services Engagements. In *Proceedings of the IEEE International Conference on Services Computing* pp 346-353.

[11] Soloway, E., Bachant, J. and Jensen, K. 1987. Assessing the maintainability of XCON-in-RIME: coping with the problems of a VERY large rule base. In *Proceedings of AAAI 87*, Morgan-Kauffman, pp 824-829

[12] Dluhy, R., and McMahon,T. 2008. Intensive Glycemic Control in the ACCORD and ADVANCE Trials. *New England Journal of Medicine.* 358(24): 2630-2633

[13] Compton, P., Kang, B., Martinez-Bejar, R., Rudrapatna, M. and Sowmya, A. 2008. Situated Cognition in the Semantic Web Era. In *Proceedings of the 2008 European Knowledge Acquisition Workshop,* Springer-Verlag, pp374-387.