Pacific Knowledge Systems

The RippleDown Advantage

"Traditional expert systems can seem easy to build and maintain, but are in fact very difficult. The RippleDown approach solves this maintenance problem"

Paul Compton
University of New South Wales
December 2012

Contents

Rules Technology

Rules based system technology is unusual in that apart from interfaces and processing speed, essentially the same techniques are used in current commercial rule products as were used in the first expert systems[1]. This is despite the fact that if one looks carefully it is clear that it has been known for many years that these systems are very difficult to build and maintain.

Matthew Fox in an article entitled "AI and expert system myths, legends and facts" summarised the state of the art in 1990.

LEGEND: AI systems are easy to maintain. Using rules as a programming language provides programmers with a high degree of program decomposability; that is, rules are separate knowledge chunks that uniquely define the context of their applicability. To the extent that we use them in this manner, we can add or remove rules independently of other rules in the system, thereby simplifying maintenance.

Building rule based systems differs from this ideal. Various problem solving methods (including iteration) require that rules implementing these methods have knowledge of other rules, which breaks the independence assumption and makes the rule base harder to maintain. The much-heralded XCON system has reached its maintainability limit (about 10,000 rules). The complexity of rule interactions at this level exceeds maintainer abilities.

Nothing has really changed. The most recent survey of knowledge based system developers by Zacharias in 2008 found In a survey of 64 developers with an average of 6.6 years of experience, 60% indicated their knowledge bases frequently gave the incorrect answer and 34% indicated that sometimes the incorrect results was given, The biggest need was identified as debugging/verification tools to correct such errors. In 2009 Zacharias wrote:

The One Rule Fallacy: Because one rule is relatively simple and because the interaction between rules is handled automatically by the inference engine, it is often assumed that a rule base as a whole is automatically simple to create...However, it is an illusion to assume that rules created in isolation will work together automatically in all situations. Rules have to be tested in their interaction with other rules on as many diverse problems as possible to have a chance for them to work in novel situations (Zacharias 2009)

The initial insight on which rule-based systems were based is that it much easier to add rules if the inference or reasoning engine is separate from the rules. That is, declarative knowledge should be separate from inference. As expressed in the Business Rules Manifesto this is still the central idea behind business rules: to separate business knowledge from data base code. It is absolutely true that this separation of inference and knowledge makes it trivially easy to add rules – the problem is whether the rules as a whole produce the desired outcomes. And as the quotes above suggest the ease of adding a few rules can be very misleading, because the problems emerge and exponentially increase as the rules multiply.

RippleDown Technology

The ideas behind RippleDown were developed initially at the Garvan Institue and then the University of New South Wales, precisely to deal with the problem of debugging a knowledge base as more and more rules were added. The Garvan had one of the first four medical expert systems to go into routine clinical use anywhere (Buchanan 1896). It was constantly maintained and the problems of debugging the rules added was very real.

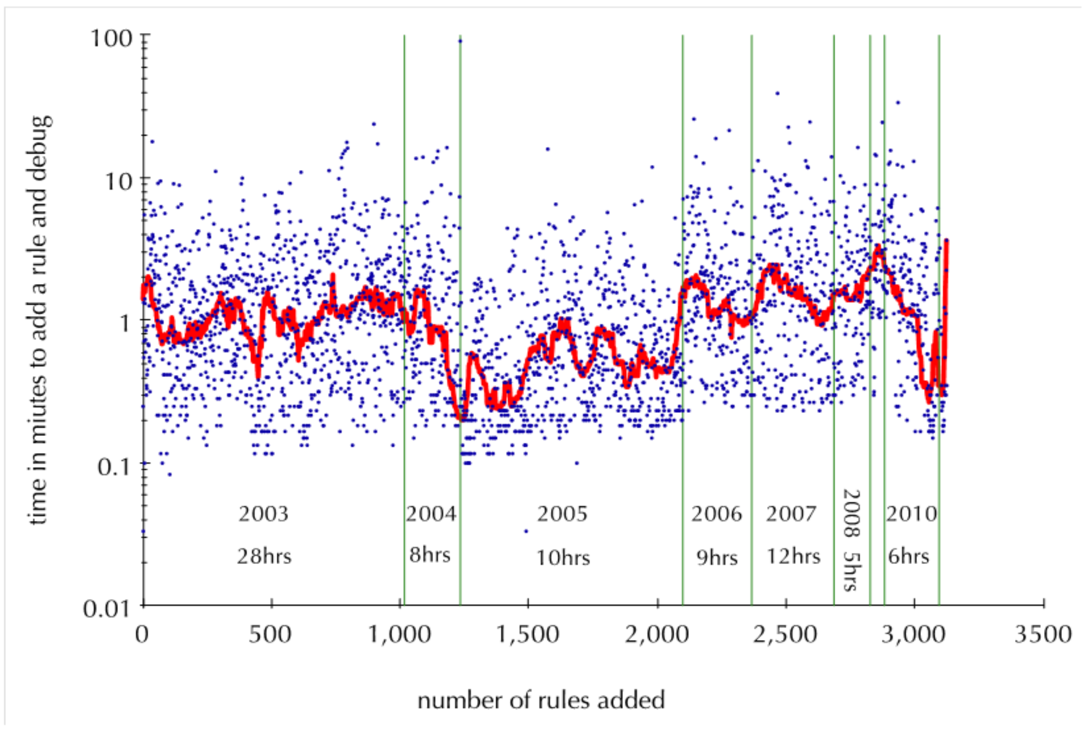
One essential idea behind RippleDown is to have different rule structure, more closely tied to the reasoning mechanism. This means that when a new rule is added, the potential interactions with other rules that cause all the debugging problems are minimised. Secondly, when these interactions do occur, rather than chasing down the rules and editing them and causing more problems, the user is simply presented with some cases, actual data, that would now be processed differently by the knowledge base and asked to accept either that the conclusion from the new rule should apply, or to select some more conditions for the rule to distinguish the cases.

At no stage does the user need to examine or explore the knowledge base. Over the years there have been a number of algorithms developed around these ideas with PKS's proprietary RippleDown technology being the most powerful. RippleDown also includes a rich language for developing rules, and PKS is happy to work with customers to add more features to this language for their domain application or develop a narrow domain specific language. As well a range of interfaces have been developed to interface with other information systems, and again PKS will work with customers to develop interfaces. Currently installations range from embedded to cloud applications.

The experience with RippleDown is summarised in the following:

- Over 170 knowledge bases are in use processing 3M cases per month
- All knowledge bases are developed by domain experts, rather than IT staff
- It takes only a minute or two to add a rule and debug a knowledge base. (This data comes from user log file; an example is shown below)
- Users add rules to knowledge bases while they are in use, as part of their normal workflow; there is no long start-up
- It remains just as rapid and easy to add a rule and debug the knowledge base after 1000s of rules are added, and years have passed.

The typical experience with adding rules and debugging the knowledge base is shown in the figure following.



This is a knowledge base with over 3,000 rules developed by a pathologist to advise GPs on the clinical significance of the biochemistry laboratory results for a patient, taking into account the patient history and whatever information is available. It is a highly sophisticated knowledge base providing over 250 different pieces of advice, which it also may combine to give overall advice.

The blue marks give the time it took to add each and rule and to carry out the related debugging and validation of the knowledge base. This is raw data; no data has been deleted or edited. The red marks represent the median time to add a rule over the last 50 rules. A log scale is used and medians shown because the time logged is the elapsed time from when the user starts to add a rule until debugging and validation is complete and they exit. The elapsed time therefore includes distractions such as telephone calls etc. The green vertical lines indicate time separated into years. The data shown as hours is the total elapsed time adding rules and debugging. Note this is the total elapsed time, including time answering the telephone.

No other technology has ever shown this ease and speed of knowledge acquisition and debugging independent of knowledge base size. And no other technology has allowed the same ease of dealing with the new cases and requirements that arise over the years.

Fox noted above, that the famous XCON system had reached its limit of maintainability with about 10,000 rules – despite a small army of knowledge engineers. The largest RippleDown system in use and still maintained has over 15,000 rules and is maintained by a single pathologist as a minor extension to their normal duties. This may be an unfair comparison because of the different domains, but nevertheless it is an illustration of how RippleDown contrasts with other rule technologies.

Applications of RippleDown

RippleDown is well established in Chemical Pathology where it is used to provide advice on laboratory results, alerts in hospitals about possible myocardial infarcts, and to assess risk factors. It is also used to check that physicians are ordering the appropriate tests to and to auto-validate laboratory results, i.e. to check whether the results can be sent out. It has been shown to have a specific impact on patient care. The technology is not in anyway specific to medicine, and can be applied to any application where rules are needed.

Rule technology is generally used where there is a high level of expertise that needs to be captured or where there are business policies that need to be expressed. Because RippleDown provides such extremely rapid and simple knowledge acquisition it is also ideal for areas of common sense where knowledge is not so easily articulated. For example, for information extraction from the Web and other unstructured data sources, it is easy for people to recognise whether they have the right information, but the rules they write tend to be very simple.

The beauty of RippleDown is that it enables these very simple rules to be very easily accumulated and refined, ending with a far more powerful text and natural language processing system than anything the user could articulate. As well as information extraction, this is an extremely powerful approach to cleansing and integrating data in big data applications.

Other Technologies

There are a range of learning technologies including neural nets, support vector machines, decision tree learners etc. If suitable data are available these should certainly be used, and in fact there is a widely used machine learning system RIDOR that uses one of the early RippleDown representations. However, it is very rare to find large enough volumes of data to enable the types of subtle concepts to be learned that RippleDown can be taught with a single case. In fact developing a RippleDown knowledge base is an ideal way of producing consistent data suitable for machine learning – if that is then still required.

References

Buchanan, B. (1986). *"Expert systems: working systems and the research literature."* Expert Systems 3: 3251.

Fox, M. (1990). *"AI and expert system myths, legends, and facts."* IEEE EXPERT Feb: 820.

Zacharias, V. (2008). *"Development and Verification of Rule Based Systems—A Survey of Developers."* RuleML 2008, Orlando, Springer-Verlag: 6-16.

Zacharias, V. (2009). *"The Debugging of Rule Bases. Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches"* IGI Global: 302&325.