Submitted to the first Goddard Workshop on Formal Approaches to Agent-Based Systems will be held 6th-7th April 2000
at NASA Goddard Space Flight Center, Greenbelt, MD, USA. http://fmw.gsfc.nasa.gov/
WP:b/99/negoiation/goddard.tex(December 6, 1999).

Agents Talking Faster

Tim Menzies‡, Bojan Cukic¶, Enrico Coiera†
‡¶NASA/WVU Software Research Lab, 100 University Drive, Fairmont, USA
†Department of Computer Science and Engineering, UNSW, Australia
`<tim@menzies.com,cukic@csee.wvu.edu,ewc@pobox.com>`

December 6, 1999

Agents running on distributed computers may interact under a variety of resource restrictions. One restriction is bandwidth. Every transmission also takes time since it must be processed by both the sender and the receiver. Studies of human communication suggests that failure to optimise the use of available resources can result in inefficient communication within organisations and consequent task errors [Coiera 1996]. How can we minimize the use of those resources for agents, and maximise the likelihood that communication tasks are successfully completed?

Consider two humans communicating via a channel with a certain bandwidth. A robust finding from human conversations is that restricting bandwidth has little impact on the effectiveness of the outcomes of many collaborative problem solving tasks [McCarthy & Monk 1994]. This occurs when the communicating individuals have a high degree of shared common ground. A particularly striking finding is that humans assume they share common world views in conversation until they detect an error, and then attempt to probe the cause of the error. Interestingly, human conversation seems to consist of both an explicit small set of exchanges devoted to testing or confirming understanding, as well as the primary exchange [H.H. Clarke 1991].

Can we use these observations to minimize the bandwidth required between computational agents? Suppose agents move through alternating stages of assumed model consensus, detection of model conflict, and then model repair. While agent models are in conflict, we broaden the bandwidth between them. When models are in consensus, we restrict the bandwidth by assuming shared knowledge between agents. As a result, bandwidth is conserved during conversation until it is explicitly demanded to reconcile the world models of conversing agents. Time is also conserved by minimising unnecessary sharing of knowledge between agents [Coiera 1999].

To apply this approach, agents have to continually test that other agents hold their beliefs. One method for doing this would be to ask agents to dump their belief sets to each other. We consider this approach impractical for three reasons:

- It incurs the penalty of the infinite regress; i.e. I believe that you believe that I believe that. . . .
- Such belief-dumps could exhaust the available band-width.

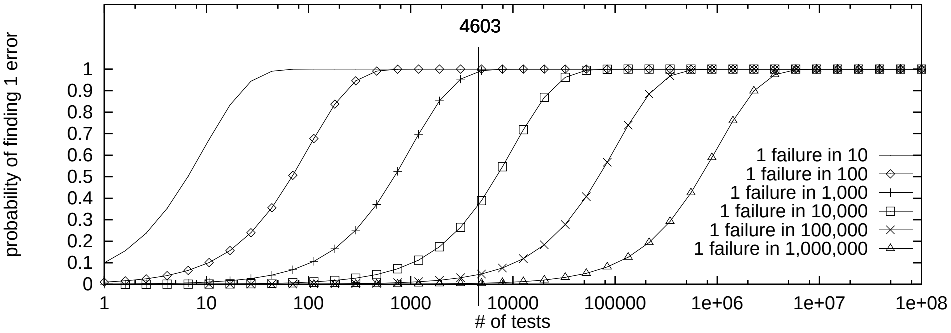

Figure 1: $C = 1 - (1 - \alpha)^N$. Theoretically, 4603 probes are required to achieve a 99% chance of detecting moderately infrequent items; i.e. ones that only occupy which $\frac{1}{1000}$-th of the model. [Hamlet & Taylor 1990].

- Agents may not wish to give other full access to their internal beliefs. For example, security issues may block an agent from one vendor accessing the beliefs of an agent from another vendor.

Without access to internal structures, how can one agent assess the contents of another? Software engineering has one answer to this question: black-box specification-based probing [Hamlet & Taylor 1990, Lowrey, Boyd & Kulkarni 1998]. Agent-A could log its own behaviour to generate a library of input-output pairs. In doing so, Agent-A is using itself as a specification of the expected behaviour of Agent-B, assuming our two agents are in consensus. If Agent-A sees those outputs when Agent-B is presented with those inputs, then Agent A could infer that Agent B has the same beliefs as itself.

Unfortunately, black-box probes can be very bandwidth expensive. Equation 1 shows a widely-used statistical model connecting the number of probes N to the confidence C that we will find an error with frequency of occurrence α.

$$C = 1 - (1 - \alpha)^N \tag{1}$$

Equation 1 says that for probes to be 99% certain of detecting anomalies, then the number of black-box probes required is 4.6 times divided by the frequency of that anomaly (see Figure 1). For example, to be 99% certain that Agent-B has less than 1% difference in its beliefs to Agent-A, then 460 probes would be required. This statistical model is hence very pessimistic on the possibility of Agent-A accurately assessing its consensus with Agent-B, without using large bandwidths for the probing.

We argue that Figure 1 is a gross over-estimate of the number of probes. Figure 1 assumes no knowledge of the internal structure of our agents. If we commit to some view of structure, then estimates can be generated for the odds of finding differences between Agent-A and Agent-B. Hence, assuming that agents are usually in consensus, then we can reduce inter-agent bandwidth as follows:

- Assume consensus and restrict bandwidth.

Figure 2: Some simulation runs for $O[h]$ (selected randomly from the 240,000 runs).

- As part of the normal inter-agent dialogue, Agent-A occasionally drops a probe question to Agent-B. If the number of probes required is very small, then these process will add little to the overall bandwidth.
- If conflicts are detected, increase the bandwidth between Agent-A and Agent-B to allow for discussions.

In our model, we assume that probing means building explanation trees across a network towards some feature of interest (i.e. we are checking that we can reach some fault or a desired feature). In this abstraction, trees are built across and-or networks towards randomly selected nodes. Networks are characterized by, amongst other things, their size, percentage of and-nodes/or-nodes, and the mean and standard deviation of the number of parents of and-nodes/or-nodes. The trees start from randomly selected inputs. The network contains inconsistent pairs; i.e. constructing this tree is an indeterminate task and is NP-hard. The results are expressed as $O[h]$; i.e. the odds that we can reach some randomly selected node using a tree of height h from some randomly selected inputs.

This model has been run 240,000 times over a wide range of parameters including network sizes, and/or-node frequencies, and number of and/or-node parents [Menzies & Cukic 1999b]. From studying the runs, we say that building the tree is like a war between the and-nodes and the or-nodes. Let oa be the ratio of or-node parents to and-node parents. At $oa < 1$ value, the and-nodes dominate and the odds of reaching a node using a tree of height h plummet away to very low levels. At $oa > 1$ value, the or-nodes dominate and the odds of reaching a node using a tree of height h rise quickly to a very high plateau (that plateau being the ratio of or-nodes in the program). Only within a very narrow range ($oa \approx 1$) does $O[h]$ neither plateau or plummet.

The values $oa < 1$, $oa{\approx}1$, $oa > 1$ gives rise to the following three types of curves:

Type #1 curves: rise to a high plateau; e.g. curve A in Figure 2. If a system has a Type #1 curve, then most parts of the program are easily probed.

Type #2 curves: plummet quickly to very low values; e.g. curve B. If a system has a Type #2 curve, then most parts of the program are unreachable. Note that, just as with curves of Type #1, we only need to test Type #2 systems with a small number of probes since a few probes will tell us as much as a very large number of probes.

Type #3: curves rise slowly to a low peak before falling away; e.g. curve C. If a system has a Type #3 curve, then some portions of the system can be reached, possible with a large number of probes.

We argue that Type #3 systems occur very rarely. Firstly, they only appear for very narrow parameter ranges. Secondly, our literature reviews of the testing field show the same pattern: in many studies, a small number of probes yields as much information as a large number of probes [Menzies & Cukic 1999*b*, Menzies & Cukic 1999*a*]. That is, contrary to Figure 1, we can assume that our agents are of type Type #1 or Type #2. Hence, the bandwidth required for agents to monitor consensus need not be excessive.

Acknowledgements

This work was partially supported by the NASA cooperative agreement #NCC 2-979.

References

Coiera, E. [1996], Communication in organisations, Technical report, Hewlett-Packard Laboratories. Technical Report HPL-96-65, May, 1996.

Coiera, E. [1999], Communication under scarcity of resources, Technical report, Hewlett-Packard Laboratories. Technical Report, 1999, to appear.

Hamlet, D. & Taylor, R. [1990], 'Partition testing does not inspire confidence', *IEEE Transactions on Software Engineering* **16**(12), 1402–1411.

H.H. Clarke, S. B. [1991], Grounding in communication, *in* 'Perspectives on Socially Shared Cognition', American Psychological Association.

Lowrey, M., Boyd, M. & Kulkarni, D. [1998], Towards a theory for integration of mathematical verification and empirical testing, *in* 'Proceedings, ASE'98: Automated Software Engineering', pp. 322–331.

McCarthy, J. C. & Monk, A. F. [1994], 'Channels, conversation, co-operation and relevance: all you wanted to know about communication but were afraid to ask', *Collaborative Computing* **1**, 35–60.

Menzies, T. & Cukic, B. [1999*a*], On the sufficiency of limited testing for knowledge based systems, *in* 'The Eleventh IEEE International Conference on Tools with Artificial Intelligence. November 9-11, 1999. Chicago IL USA.'. Available from `http://research.ivv.nasa.gov/docs/techreports/1999/NASA-IVV-99-011.pdf`.

Menzies, T. & Cukic, B. [1999*b*], 'When to test less', *IEEE Software (submitted)* .